Université François Rabelais Tours
Ecole Doctorale : Santé, Sciences et Technologies
Année Universitaire : 2006-2007

THESE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITE DE TOURS

Discipline : **Informatique**

Présentée et soutenue publiquement

par :

Sylvain Hocquet

le

Authentification biométrique adaptative
Application à la dynamique de frappe et à la signature manuscrite

---

Sous la direction de Hubert Cardot et Jean-Yves Ramel

Jury

CARDOT Hubert	Directeur de thèse	Professeur, Université François Rabelais Tours
CHOLLET Gérard	Rapporteur	Directeur de Recherche CNRS, GET-ENST Paris
GLOTIN Hervé	Examinateur	Maître de conférences, Université du Sud (Toulon)
LOPEZ-KRAHE Jaime	Rapporteur	Professeur, Université de Paris 8
RAMEL Jean-Yves	Examinateur	Maître de conférences HDR**,** Université François Rabelais Tours
SLIMANE Mohamed	Examinateur	Professeur, Université François Rabelais Tours

Remerciements

Tout d'abord, je tiens à remercier mon directeur de thèse, Hubert Cardot ainsi que Jean-Yves Ramel co-encadrant, pour leurs conseils, le temps précieux qu'ils m'ont accordé et pour leur soutien au cours de ces années de travail. Ce soutien sans faille et leurs réflexions critiques sur mes propositions, m'ont permis de mener à bien ces travaux.

Je voudrais ensuite remercier le groupe CapMonétique en général, et ses membres en particulier, pour m'avoir proposé ce sujet de thèse et m'avoir accueilli dans leur entreprise. Ils m'ont permis de garder un ancrage au réel lors de mes travaux de recherche, de tester en situation mes travaux, et ont participé activement à l'évaluation de mes propositions.

Je remercie également tous les membres du laboratoire d'informatique, et plus particulièrement de l'équipe RFAI, qui m'a accueilli pour ces années de thèse.

Je remercie le DI de l'école polytechnique universitaire de Tours, pour m'avoir permis de travailler et d'encadrer des étudiants dans le cadre de leur projet de fin d'étude. Je remercie bien sûr tous ces étudiants : Laurent, Jérémy, Pascal, Xavier, Bertrand et Antoine qui ont travaillé avec moi et m'ont chacun apporté une aide précieuse dans mes travaux.

Je dis un grand merci à tous les anonymes qui m'ont permis de constituer la base de données utilisée par mes travaux, c'est-à-dire les bases de test que j'ai utilisées, et je m'excuse de tous les désagréments que je leur ai infligés pour cela (plantage, perte de données, crampe à la main...).

Je voudrais remercier mes parents, mon frère et ma sœur, qui m'ont soutenu au cours de cette thèse et qui n'ont jamais hésité à payer de leur personne en jouant les bêta-testeurs de mon application.

Je terminerais en remerciant Clo, qui a dû me subir pendant la rédaction de ce manuscrit, et qui m'a apporté réconfort et soutien.

SOMMAIRE

Table des Figures

Table des Tableaux

Résumé

La biométrie comportementale étudie les comportements des individus plutôt que leurs caractéristiques physiques. De par sa nature, l'implémentation d'un système utilisant la biométrie comportementale entraîne souvent la résolution de problèmes à une classe, c'est-à-dire de problèmes dans lesquels il n'est possible d'utiliser que des informations provenant d'individus appartenant tous à une unique classe. L'objectif est alors de construire un système capable de distinguer les individus appartenant à cette même classe des autres individus (utilisateurs inconnus considérés comme des imposteurs).

Située dans ce contexte, la première partie de cette thèse passe en revue les principaux objectifs et travaux réalisés dans le cadre de la classification à une classe et dans le cadre de la biométrie afin d'en dégager les verrous scientifiques qui restent à résoudre.

La seconde partie de ce manuscrit décrit et justifie les différentes techniques et méthodes que nous proposons pour la mise en place d'un système d'authentification biométrique utilisant des caractéristiques comportementales. La plupart de nos propositions se basent sur l'utilisation d'une base de référence afin de faire en sorte que le système s'adapte au maximum et automatiquement à chaque utilisateur notamment par la détermination de paramètres personnalisés.

La dernière partie de ce manuscrit présente des mises en application de l'architecture proposées et de nos préconisations dans le cadre d'une application d'analyse de la dynamique de frappe au clavier et d'une application d'analyse de signatures manuscrites. La première application a été demandée par la société CAPMONETIQUE qui a financé ce travail. La dynamique de frappe est une méthode biométrique qui connaît un important développement. Cette méthode étudie l'interaction des utilisateurs avec leurs claviers afin de les authentifier. Cette partie, ainsi que les expérimentations effectuées sur l'authentification de signatures manuscrites démontrent l'intérêt de nos préconisations et valident la généricité puisque, dans les deux cas les performances des systèmes biométriques comportementaux augmentent de manière significative.

Introduction

Depuis le 11 septembre 2001, le monde connaît une demande toujours plus importante en termes de sécurité. Au départ, cette demande s'exprimait à l'échelle des états : contrôles aux frontières, luttes anti-terroriste, contrôles de l'immigration… Mais, cette exigence de sécurité s'est vite étendue à la vie de tous les jours. Au sein des entreprises d'abord, où les efforts se concentrent sur la mise en place de dispositifs de contrôle d'accès, sur la surveillance des salariés et sur la sécurisation des réseaux internes à l'entreprise. Puis aujourd'hui, l'individu même s'en préoccupe et désire mieux protéger son domicile, sa voiture, et l'accès à son ordinateur personnel qui contient de plus en plus de données essentielles (carnets d'adresses, courriels importants, codes de cartes bancaires…). La protection des données personnelles informatiques est d'ailleurs une problématique critique pour l'avenir du développement d'Internet, que ce soit pour le commerce électronique, les courriels ou les autres applications en ligne... Mais, malgré cette nécessité vitale de sécurité, beaucoup d'entreprises ou de particuliers ne sont pas prêts à payer le prix de cette sécurité, prix qui est bien sûr financier mais qui s'évalue aussi en effort à fournir pour assurer cette sécurité.

Pour répondre à ces nouveaux besoins, la biométrie semble être une solution pratique, efficace et dont le coût en effort et en argent est en constante diminution. De fait, la biométrie connaît un développement fulgurant. Cet engouement entraîne le développement de méthodes biométriques très variées : des plus classiques, comme l'étude des empreintes digitales [Jain *et al.*, 1997] ou de l'iris [Tisse, 2003], aux plus exotiques comme la reconnaissance de la démarche [Yam *et al.*, 2002], la reconnaissance de la forme de l'oreille [Yan et Bowyer, 2005]. Les industriels proposent de plus en plus, pour les problèmes exigeant énormément de sécurité, de ne plus utiliser une seule caractéristique mais de mettre en place un système basé sur des combinaisons de différents moyens biométriques afin d'accroître encore la sécurité.

Malgré ce développement rapide, la biométrie comporte des points d'imperfections. En effet, actuellement, il y a encore bien souvent trop peu de réflexions avant l'implémentation d'une solution biométrique, que ce soit au niveau de la méthode choisie, des contraintes imposées aux usagers ou du niveau de sécurité

choisi. Il y a ainsi parfois des aberrations : capteurs très performants couplés à des algorithmes de reconnaissance obsolètes, le tout pour contrôler l'accès à un restaurant scolaire.

Outre ce manque de réflexion préalable, l'autre point critique des systèmes biométriques concerne leur fiabilité et les mécanismes de reconnaissance ou d'authentification à mettre en œuvre. C'est sur ce point que porte notre travail. Nos principaux efforts ont visé à identifier les verrous actuels des systèmes de biométrie comportementale et à proposer des solutions simples pour les résoudre.

L'authentification biométrique peut souvent être considérée comme un problème de classification à une classe puisque lors de la conception du système et lors de la reconnaissance d'un individu, le système ne possède que très peu d'informations pour prendre sa décision (données provenant uniquement d'un individu).

Le problème à une classe ne se limite pas à l'authentification biométrique, il se retrouve dans plusieurs autres domaines comme par exemple dans la détection de la fausse monnaie [He *et al.*, 2004], la classification de documents [Manevitz et Yousef, 2002] , la détection de pannes [Tax *et al.*, 1999], …

Les objectifs de cette thèse, réalisée en partenariat avec le groupe Capmonétique, est aussi d'étudier et de faire de nouvelles propositions concernant les méthodes, outils et systèmes pouvant être utilisés pour produire de tels classificateurs et notamment d'étudier la mise en place d'un système biométrique utilisant la dynamique de frappe au clavier.

Les études menées nous ont permis d'émettre des recommandations et des procédures à suivre. Ces recommandations et procédures sont présentées sous la forme d'une architecture suffisamment générique pour être mise en place dans de nombreuses applications. Le fait de se retrouver dans un problème à une classe entraîne des répercussions et des contraintes sur tous les modules du système biométrique, de l'acquisition, l'enregistrement, la normalisation des données jusqu'à la reconnaissance.

Ce manuscrit comporte trois parties. Dans la première partie, nous présentons les principaux objectifs et les travaux effectués dans le domaine de la classification à une classe et de la biométrie afin d'en dégager les verrous scientifiques et les études qui restent encore à mener actuellement.

Dans la seconde partie du manuscrit, nous décrivons l'architecture que nous préconisons pour la mise en place d'un système d'authentification biométrique à partir de caractéristiques comportementales. A partir de cette architecture, nous décrivons et justifions les différentes techniques et méthodes à implanter dans un tel système. La plupart des améliorations que nous proposons se base sur l'utilisation d'une base de référence afin de faire en sorte que le système s'adapte au maximum et automatiquement à chaque utilisateur notamment par la détermination de paramètres qui lui sont spécifiques.

La dernière partie de ce manuscrit présente une mise en application de cette architecture et de nos préconisations dans le cadre de deux applications. La première d'entre elle est une application d'analyse de la dynamique de frappe au clavier. Cette application est demandée par la société CAPMONETIQUE qui a financé ce travail. La dynamique de frappe est une méthode biométrique qui connaît un important développement. Cette méthode étudie l'interaction des utilisateurs avec leurs claviers afin de les authentifier.

La seconde concerne l'adaptation de notre architecture à un problème de reconnaissance de signatures manuscrites en ligne. Dans cette partie, nous démontrons l'intérêt de suivre nos recommandations et l'architecture que nous proposons.

Ces deux expériences nous permettent de conclure, en montrant l'intérêt de nos propositions.

Chapitre 1.
L’authentification biométrique

Aujourd'hui, la biométrie est préconisée comme une des solutions les plus intéressantes pour résoudre la plupart des problèmes de sécurité. Néanmoins, il demeure de très nombreux problèmes et incertitudes quand à son efficacité ou son utilisabilité à grande échelle. Dans ce chapitre, nous commençons par revenir brièvement sur les principaux concepts liés à la biométrie ainsi que ses limites. Nous présentons ensuite les techniques actuelles d'authentification biométrique, les contraintes qui y sont associées et procédés utilisés actuellement.

1.1. La biométrie

1.1.1. Définition

Lorsque l'objectif est de limiter l'accès à une ressource ou du moins de savoir qui y accède, il existe trois grands moyens qui peuvent éventuellement être combinés :

- Utiliser ce que l'on a (carte magnétique, badge…)
- Utiliser ce que l'on sait (mot de passe, code PIN…)
- Utiliser ce que l'on est (biométrie)

La biométrie se base sur des caractéristiques liées à chaque individu. Ces caractéristiques peuvent appartenir à deux grandes familles. Les premières d'entre elles, qui sont également les plus utilisées, correspondent aux traits physiques des usagers (empreinte digitale, iris…). La seconde famille qui reste encore assez peu utilisée mais dont l'usage à tendance à se développer correspond aux caractéristiques comportementales (signature manuscrite, dynamique de frappe au clavier, reconnaissance de la voix…).

Le grand public a pris conscience du développement de la biométrie et de son utilisation dans la vie de tous les jours à travers de nombreux films qui ont ainsi contribués à mettre ce marché sur le devant de la scène. En fait, la biométrie est utilisée depuis longtemps, notamment par les services policiers ou judiciaires chez qui l'utilisation des empreintes digitales s'est généralisée depuis la fin du 19^{e} siècle et le système Bertillon. Aujourd'hui, les évolutions de la biométrie portent, d'une part, sur l'automatisation des traitements : l'ordinateur doit remplacer les experts humains, et d'autre part, sur la mise en place de fichiers de caractéristiques

biométriques à grande échelle. De nos jours, les entreprises et même les particuliers commencent à avoir accès à cette technologie à des prix de plus en plus bas.

1.1.2. Différentes méthodes biométriques

Les systèmes biométriques n'utilisent pas tous la même méthode pour différencier les individus. Il existe en fait autant de méthodes qu'il y a de caractères physiques et comportementaux permettant de comparer deux individus. Néanmoins pour pouvoir être utilisée, une caractéristique biométrique doit normalement vérifier les conditions suivantes :

- Elle doit être unique pour chaque individu
- Elle doit être constante au cours du temps
- Elle ne doit pas être copiable

De nombreuses caractéristiques biométriques, comme la dynamique de frappe au clavier ou la reconnaissance de signatures manuscrites, ne vérifient pas toutes ces conditions. Par exemple, la façon de taper au clavier d'une personne évolue naturellement au cours du temps suivant le degré d'entraînement de la personne, suivant son âge… Mais, cette évolution est lente (voire très lente) et peut donc être considérée comme négligeable entre deux utilisations du système. De même, la contrainte d'unicité peut être assouplie en considérant que la probabilité que deux personnes prises au hasard ait la même valeur pour une caractéristique donnée est quasiment nulle, c'est-à-dire inférieure à un certain seuil dépendant de l'application prévue. Malgré les assertions des industriels, la plupart des caractéristiques biométriques sont copiables. Par exemple, dans [Matsumoto *et al.*, 2002], les auteurs montrent la facilité de tromper des capteurs d'empreintes digitales du commerce à l'aide de faux doigts. En fait, les décideurs utilisent quand même la biométrie car ils considèrent que les efforts et les investissements nécessaires pour copier une donnée biométrique suffisent généralement à dissuader d'éventuels contrefacteurs.

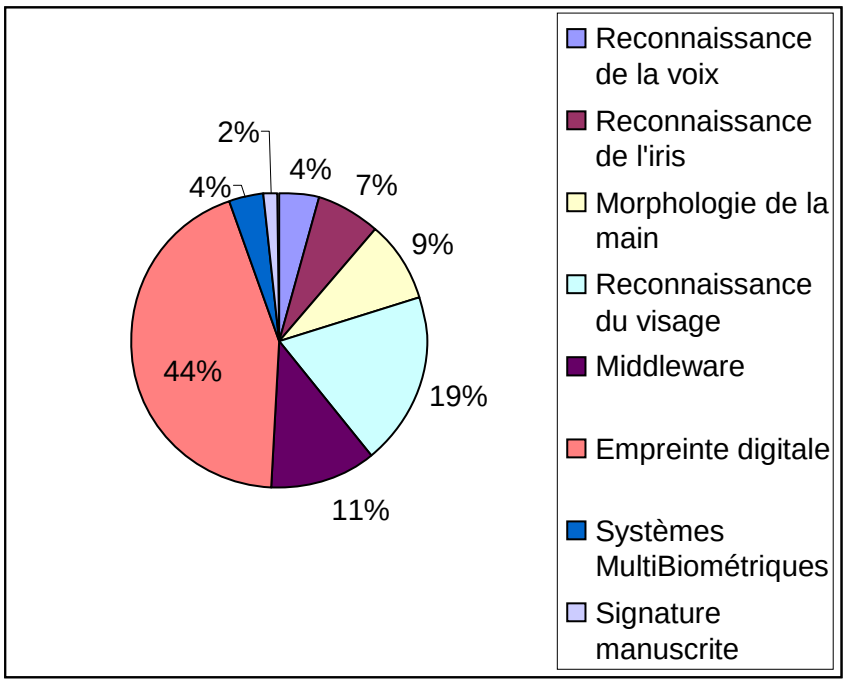


Figure 1 : Parts de marché des techniques biométriques en 2003 [IBG]

La Figure 1 réalisée d'après les chiffres de l'International Biometric Group [IBG] montre les parts de marché des principales méthodes biométriques en 2003. Encore aujourd'hui, ce sont les empreintes digitales qui sont les plus utilisées, suivies par la reconnaissance faciale. Ces deux méthodes représentent, à elles seules, les deux tiers du marché de la biométrie. En fait, il est généralement admis qu'à un problème donné, correspond une solution biométrique particulière. Par exemple, pour le contrôle de l'accès à un bâtiment ultra sécurisé pour lequel le coût de protection n'a pas grande importance, un scanner rétinien peut être utilisé. Par contre, la protection d'un photocopieur se fera plus simplement par un système à base d'empreintes digitales qui coûte beaucoup moins cher et qui offre d'assez bonnes performances. Sur la Figure 2, réalisée également par [IBG] sont présentés les avantages et inconvénients des principales méthodes biométriques.

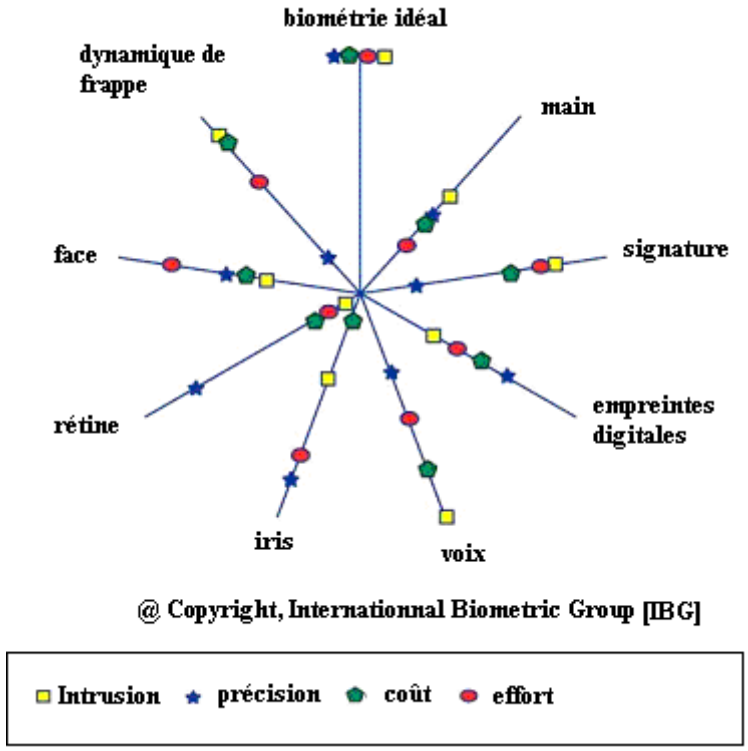


Figure 2 : Comparaison des différentes méthodes biométriques

Sur ce schéma, les différentes méthodes sont évaluées à l'aide d'une série de critères :

- Effort : effort fourni par l'utilisateur lors de l'authentification
- Intrusion : information sur l'acceptation du système par les usagers
- Coût : coût de la technologie (lecteurs, capteurs, etc.)
- Précision : efficacité de la méthode (liée au taux d'erreur)

La Figure 2 montre qu'il n'existe pas de méthode idéale. Les méthodes se divisent en deux grands groupes. Le premier groupe englobe les méthodes conviviales pour les utilisateurs (effort à fournir faible, méthode peu intrusive, prix modéré) mais assez peu performantes. Ce groupe qui correspond aux méthodes basées sur la biométrie comportementale (reconnaissance de la voix, de la signature…). L'autre groupe contient les méthodes plus sûres (méthodes intrusives et prix élevés, performances très bonnes). Il est donc nécessaire de déterminer, au cas par cas, pour chaque problème, la méthode qui conviendra le mieux à la situation. Pour cela, il faut étudier attentivement le niveau d'exigence en sécurité, le budget

pouvant être investi dans le système et la façon dont risque de réagir les utilisateurs. Actuellement, pour la mise en place des grands projets de passeports biométriques, les systèmes retenus par l'Europe semble être un stockage de la photo d'identité, des empreintes digitales et de l'iris sous forme numérique. A noter que le choix du ou des dispositifs biométriques peut aussi dépendre de la culture locale. Ainsi en Asie, les méthodes nécessitant un contact physique comme les empreintes digitales sont rejetées pour des raisons d'hygiène alors que les méthodes basées sur l'iris sont très bien acceptées. La dynamique de frappe fait partie des méthodes biométriques les moins performantes mais très intéressantes au niveau du coût, de l'effort à fournir et du niveau d'intrusion perçu. Elle est donc adaptée aux applications de sécurisation des zones peu sensibles et pour lesquelles il n'y a pas la volonté ou la possibilité de débloquer des budgets très élevés.

1.1.3. Précautions d'utilisation

1.1.3.1. Menace pesant sur la vie privée

En France, le stockage de données biométriques est assimilé à la constitution de bases de données personnelles. Par conséquent, toute base de données biométriques est soumise à l'approbation de la Commission Nationale de l'Informatique et des Libertés (CNIL) [CNIL]. La CNIL est un organisme indépendant chargé par la loi n° 78-17 du 6 janvier 1978, de contrôler la gestion de tous les fichiers de données personnelles. Son domaine d'activité a été étendu aux traitements automatisés des données, puis naturellement à la biométrie. Le développement de bases de données biométriques est aujourd'hui une de ses grandes préoccupations. Ainsi, les méthodes biométriques sont très surveillées en France, notamment celles basées sur les empreintes digitales. Au cours de l'année 2004, la CNIL a rendu deux décisions importantes dans ce domaine. La première (n°04-017 du 8 avril 2004) autorise la mise en place d'un système de contrôle d'accès, par l'empreinte digitale, aux zones sécurisées des aéroports d'Orly et de Roissy à la condition que les données soient stockées dans une carte à puce. La deuxième (n°04-018 du 8 avril 2004), rend un avis défavorable à la mise en place d'un système de contrôle des horaires de travail par empreintes digitales. Dans ce dernier cas, les

données devaient être stockées dans une base de données centralisée. Deux motivations expliquent la différence entre ces deux décisions :

- Tout d'abord, la CNIL demande de privilégier le stockage des données biométriques sur un support individuel pour faire face au risque de détournement de ces données. En effet, le stockage dans une base de données centralisée, comporte des risques importants : une attaque réussie ou une panne sur la base de données peut entraîner une mise hors service du système voire la compromission des données biométriques de tous les utilisateurs. Le stockage des données dans un support personnel (une carte à puce) réduit les risques, mais n'est pas exempt de tous défauts : le support peut en effet être perdu, volé, prêté ou pire dupliqué…
- Ensuite, la CNIL reste défavorable à l'utilisation des empreintes digitales en l'absence d'un impératif de sécurité incontestable, du fait de l'existence de grandes bases de données dans le domaine policier et de la facilité de « voler » une empreinte à l'insu de son propriétaire.

Ainsi, le facteur déterminant dans les décisions de la CNIL est le rapport entre le moyen utilisé, les risques engagés concernant la protection de la vie privée comparés aux impératifs de sécurité. La CNIL fait ainsi la différence entre données biométriques sensibles, c'est-à-dire que nous laissons dernière nous dans la vie courante (ADN, empreintes digitales…), et données biométriques non sensibles qui nécessitent la coopération de la personne pour être acquises (morphologie de la main par exemple).

Enfin la CNIL demande d'informer correctement les personnes qui utiliseront le système biométrique sur les modalités de contrôle et de stockage et aussi sur les motifs et les objectifs ayant conduit à l'implantation du système biométrique. Les contraintes que fait peser la CNIL sur la conception de système biométrique ne peut être ignoré dans la conception d'un système, si on veut que celui-ci soit utilisable.

1.1.3.2. La fiabilité parfaite n'existe pas

Quand on s'intéresse à la biométrie, il faut être conscient de deux choses : Premièrement, malgré les progrès techniques, la biométrie n'est pas fiable à 100%. Tout système biométrique aussi sophistiqué soit-il, admet une marge d'erreur. Il faut donc jouer sur le niveau de sécurité afin de trouver un compromis entre les contraintes imposées aux utilisateurs et l'exigence de sécurité. Souvent, les industriels ont tendance à utiliser la biométrie seule, sans autre considération de sécurité. Cela revient à mettre une serrure très compliquée et inviolable sur une porte branlante qui peut être renversée d'une simple poussée. Il faut être conscient qu'une attaque à distance sur un serveur mal protégé peut permettre de voler les profils, et une fois l'information biométrique récupérée, il est possible de la copier et de l'utiliser autant de fois que nécessaire. De plus, il est impossible de changer la plupart des identifiants biométriques alors qu'un mot de passe peut être modifié si besoin. Il est donc souvent préférable de coupler plusieurs méthodes, comme par exemple identifiant, mot de passe et biométrie. Il est même possible de coupler plusieurs méthodes biométriques afin d'obtenir une méthode encore plus fiable.

1.1.4. Fonctionnement d'un système biométrique

L'étude du fonctionnement d'un système biométrique, que nous conduisons ici permet de mettre en évidence quelques points essentiels de la mise en place d'un système biométrique. Quel que soit le système biométrique mis en place, celui-ci comporte toujours deux briques principales : le module d'enregistrement ou d'enrôlement et le module de reconnaissance.

1.1.4.1. Phase d'enregistrement/enrôlement

La phase d'enregistrement qui peut aussi être appelée phase d'enrôlement (Figure 3) consiste en l'enregistrement des caractéristiques biométriques d'un utilisateur sur un support de stockage. L'utilisateur fournit, au cours de cette phase, un ou plusieurs échantillons de la donnée biométrique utilisée (empreintes digitales, images de son iris ou de sa rétine, quelques prototypes de sa signature...). L'acquisition de multiples exemplaires d'une même donnée biométrique se justifie par la variabilité inévitable due aux nombreux facteurs pouvant influencer l'acquisition. L'ensemble des informations obtenu est stocké, dans ce qui est appelé le profil de l'utilisateur.

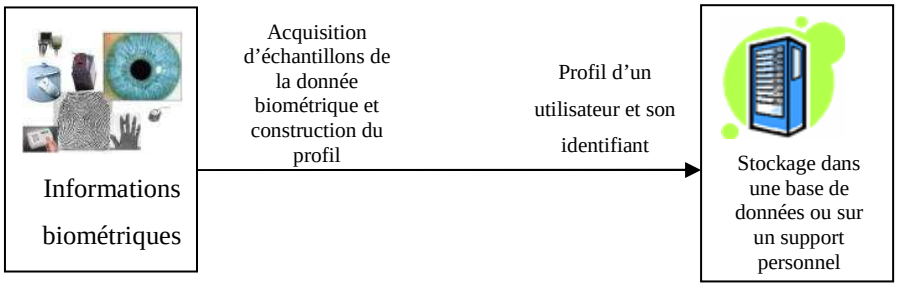


Figure 3 : Module d'enregistrement classique d'un système biométrique

La première étape de l'enregistrement est l'acquisition des données biométriques. Les problèmes posés par cette étape sont nombreux et proviennent aussi bien du système d'acquisition que de l'utilisateur lui-même. Lors de la

conception du système biométrique, une grande attention doit être apportée aux données biométriques choisies et aux capteurs utilisés pour les numériser. La fiabilité et la précision de ces capteurs influencent considérablement le fonctionnement du système. En cas de panne ou de données imprécises, le système biométrique ne fonctionnera pas ou du moins pas comme prévu. L'autre source de problèmes vient de l'environnement extérieur et notamment de l'utilisateur lui-même. Le lieu d'acquisition joue un rôle dans la précision des données obtenues.

L'utilisateur est l'élément extérieur du système sur lequel nous avons le moins de prise. En effet, quelles que soient les données biométriques choisies, un effort important lui est demandé pour fournir son identifiant biométrique. Les performances du système dépendent en grande partie de la qualité de l'effort fourni. Cet effort peut être particulièrement important surtout au cours de la phase d'enregistrement. Plusieurs facteurs peuvent influencer la qualité de l'information fournit par l'utilisateur : il peut avoir été dérangé pendant l'acquisition, être fatigué, avoir mal compris les consignes, voire même volontairement faire échouer le processus pour entraîner un dysfonctionnement du système.

La deuxième partie de la phase d'enregistrement consiste en la construction du profil de l'utilisateur. Ce profil regroupe l'ensemble des données qui seront stockées et qui serviront lors des phases de reconnaissance. Quel que soit le type de caractéristiques biométriques choisies, une réflexion sur la manière d'encoder les données sur le support de stockage doit être menée. Le profil d'un utilisateur peut contenir différentes informations pour permettre sa reconnaissance : la totalité des données brutes issues des capteurs peut être stockée mais il est souvent plus intéressant de ne stocker que les données nécessaires à la reconnaissance.

1.1.4.2. Phase de reconnaissance

Ce second module du système biométrique permet d'effectuer la reconnaissance des individus. C'est au cours de cette phase qu'une décision sera prise concernant l'identité de l'utilisateur. Cette phase diffère en fonction de l'objectif recherché : Veut-on authentifier ou identifier un utilisateur ?

L'identification (Figure 4) est la détermination de l'identité d'un individu inconnu qui se présente devant le dispositif biométrique. Le système capte ses caractéristiques biométriques et les compare ensuite avec l'ensemble des profils stockés dans la base de données contenant les informations biométriques d'une

population. Le système peut alors soit attribuer à l'individu inconnu l'identité correspondant au profil le plus proche retrouvé dans la base, soit rejeter l'individu. Dans le cas de l'identification, le système dispose et utilise les profils de tous les utilisateurs enregistrés au préalable pour prendre sa décision. Durant la reconnaissance, le système effectue une comparaison de l'identifiant biométrique d'un individu avec tous les profils stockés dans la base, on parle de test 1 : N. Le système de reconnaissance dispose alors d'une quantité importante d'informations issue des profils de tous les utilisateurs pour réaliser son travail. Il est important d'ajouter dans un système d'identification la possibilité de rejet afin de pouvoir interdire l'accès à un utilisateur si les données biométriques qu'il a fournies sont trop différentes des profils stockés dans la base. Cette possibilité est importante notamment en cas d'intrusion d'un utilisateur qui ne figure pas dans la base des profils. Une usurpation d'identité par quelqu'un d'inconnu peut ainsi être évitée.

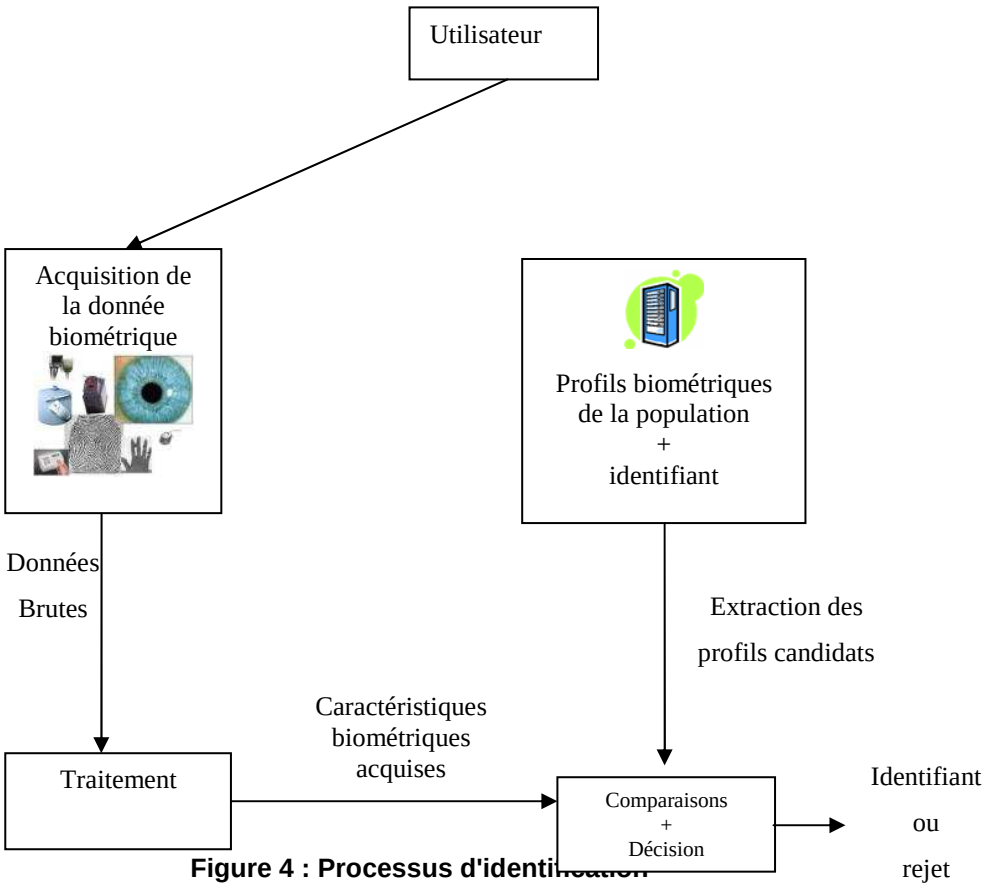


Figure 4 : Processus d'identification

L'authentification (Figure 5) est un problème différent : l'utilisateur indique son identité au moyen d'une carte ou d'un login « Je suis M. Dupont » et le système

doit alors vérifier si l'utilisateur dit vrai. Pour cela, il compare l'identifiant biométrique acquis avec le profil de M. Dupont obtenu au préalable durant la phase d'enregistrement. Le système de reconnaissance ne dispose alors que des données acquises et du profil de M. Dupont pour prendre une décision, on parle alors de test 1 : 1. Le système renvoi uniquement une décision OUI/NON.

Il arrive fréquemment qu'on utilise un système d'identification pour résoudre un problème d'authentification. L'utilisateur fournit alors un identifiant et ses données biométriques. Si le profil le plus proche du profil acquis correspond à l'identifiant fourni alors la personne est considérée comme identifiée.

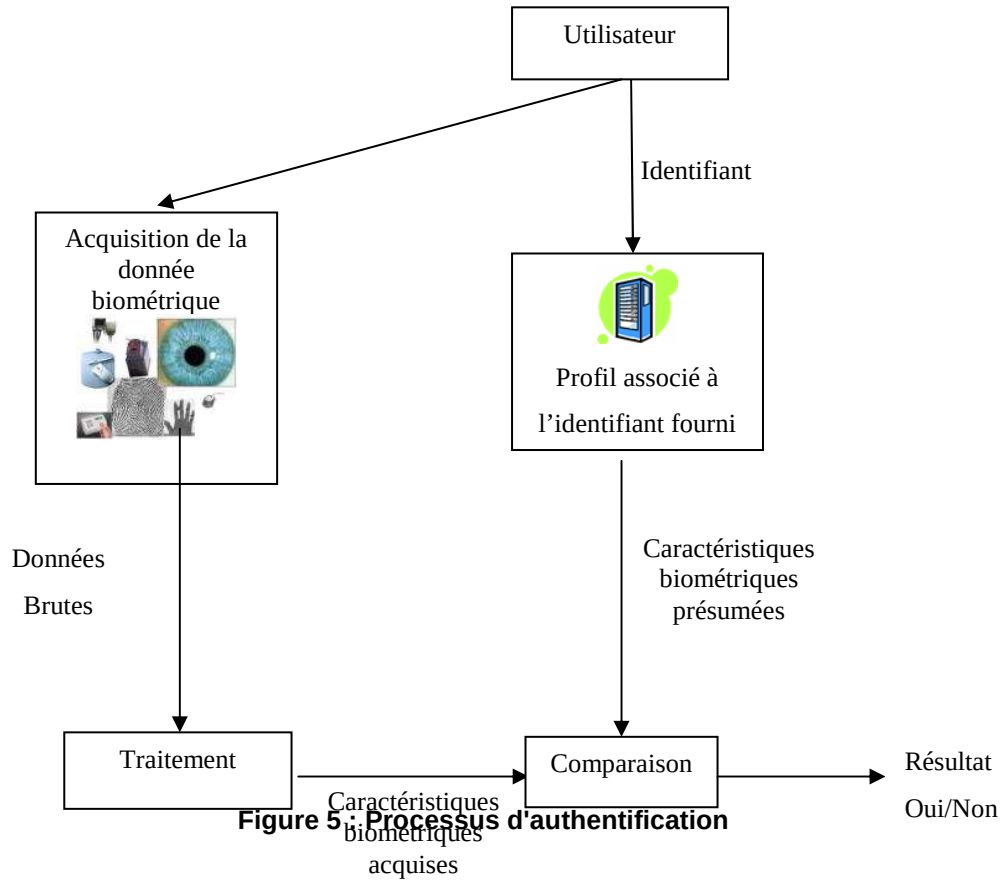


Figure 5 : Processus d'authentification

Lors de la phase de reconnaissance, l'acquisition des données biométriques demande moins d'effort de la part de l'utilisateur puisque ce dernier n'a à fournir qu'un seul échantillon de son identifiant biométrique. Les contraintes imposées à l'utilisateur doivent néanmoins rester minimes, surtout s'il est prévu que la phase de reconnaissance soit fréquemment répétée. La comparaison des données fournies par l'utilisateur avec les données contenues dans un profil se doit d'être rapide. Cette

étape est le cœur du système biométrique, elle détermine les performances du système.

Chacune des étapes peut être sujette à des erreurs ou attaques, qu'elles soient volontaires ou non de la part des utilisateurs [Faundez-Zanuy, 2004]. Il faut donc travailler et évaluer chacune si l'on veut garantir les performances d'un système biométrique.

1.1.5. Mesure de performances

Comme nous l'avons déjà indiqué, dans la pratique aucun système biométrique ne peut être entièrement fiable. Il est donc nécessaire de prévoir différents moyens de mesurer les performances des différentes techniques. En effet, la mesure des performances d'un système biométrique n'est pas triviale. Les caractéristiques d'un système mises en avant sont souvent les indicateurs de performance de la phase de reconnaissance. Ceux-ci sont les plus parlants mais ils diffèrent selon le type de système : un problème d'authentification ou d'identification [NSTC, 2006].

L'indicateur de performance utilisé pour un problème d'identification est le Taux de personne Bien Classée (T.B.C.). Pour comparer la performance de méthodes biométriques, on peut également utiliser la courbe CMC (*Cumulative Match Characteristics*), qui indique pour un entier *n* la probabilité que le système retourne le bon identifiant pour une observation dans les *n* premières réponses fournies par le système d'identification.

Pour évaluer les systèmes d'authentification, plusieurs mesures d'erreurs sont nécessaires. En effet, différents cas peuvent se produire lors de la phase de reconnaissance (rejet/acceptation) :

- les utilisateurs peuvent être acceptés de façon justifiée, quantifiés par le TVA (Taux de Vraie Acceptation)
- les imposteurs peuvent être acceptés par erreur, quantifiés par le TFA (Taux de Fausse Acceptation)
- les utilisateurs peuvent être rejetés à tord, on parle alors de TFR (Taux de Faux Rejet)
- les imposteurs peuvent être rejetés, on parle de TVR (Taux de Vrai Rejet)

Souvent l'évaluation de la performance de la méthode est réalisée à l'aide des deux situations d'erreurs de classement, c'est-à-dire en utilisant le TFA et le TFR. Ces deux taux sont liés. En jouant sur les paramètres du système, notamment en faisant varier le seuil de sécurité, ils sont modifiés de façon importante. Ainsi le choix du niveau de TFA et de TFR dépend de l'application étudiée et du niveau de sécurité souhaitée. Si, par exemple, les concepteurs souhaitent une relative facilité d'utilisation c'est-à-dire, s'ils ne désirent pas que l'utilisateur recommence plusieurs fois le processus d'authentification (surtout s'il est contraignant), on choisira un niveau de sécurité correspondant à un faible TFR, et donc avec une erreur plus grande en TFA ce qui pourrait permettre à quelques imposteurs de tromper le système . A l'inverse, s'ils désirent un système ultra sécurisé, les concepteurs imposeront alors un seuil de sécurité correspondant à un TFA proche de zéro ce qui peut faire monter le TFR aux alentours de 20 % ou plus. Le risque est alors d'empêcher une partie des utilisateurs authentiques d'accéder à la zone sécurisée. Pour comparer de façon plus simple les méthodes d'authentification, les chercheurs utilisent fréquemment le Taux d'Erreur Egale (TEE) qui correspond à la valeur pour laquelle les deux taux d'erreur sont identiques (Figure 6).

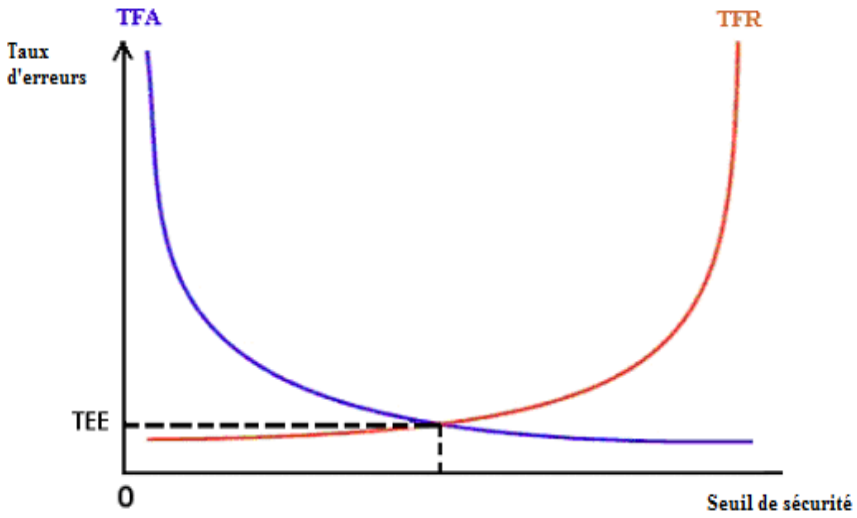


Figure 6 : TFR, TFA et TEE

Pour comparer les performances de deux systèmes biométriques il est aussi possible d'utiliser une représentation graphique appelée courbe ROC (*Receiver Operating Characteristic curve)* [Mason et Graham, 2002] montrant l'évolution d'un des taux en fonction de l'autre (Figure 7). Sur cette courbe chaque point a pour ordonnée le TFA et pour abscisse le TFR. Cette courbe est obtenue en faisant varier le seuil de sécurité sur une plage de valeurs prédéfinie. La comparaison de deux systèmes de performance se fait ensuite en comparant l'aire sous la courbe ROC des deux systèmes.

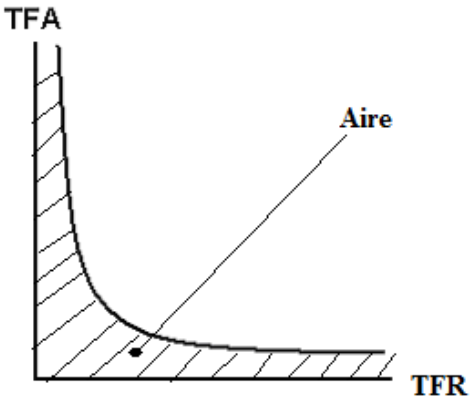


Figure 7 : Courbe ROC

L'utilisation unique des taux de reconnaissance définis ci-dessous pour comparer deux systèmes, peut être trompeuse. Les conditions expérimentales ont une influence très grande sur ces derniers. Dans [Jonathon Phillips *et al.*, 2000], les

auteurs reviennent en détail sur la façon de comparer deux systèmes biométriques et sur le problème de la définition d'un protocole de tests rigoureux. Ils mentionnent notamment la taille des bases de test et les conditions expérimentales qui peuvent affecter très sérieusement le calcul des taux d'erreur. Les auteurs montrent que dans le cadre d'un problème de reconnaissance de la voix par téléphone, le TFR est fortement augmenté quand des combinés téléphoniques différents sont utilisés, ce qui montre l'importance de bien déterminer et préciser les conditions expérimentales de chaque test.

Les taux de performance du système de décision ne suffisent donc pas à comparer deux systèmes. Dans [Peacock *et al.*, 2004] une réflexion est menée sur des critères complémentaires pouvant être utilisés pour comparer deux méthodes d'authentification. Ils concernent les coûts d'enregistrement et de reconnaissance :

- Le coût d'enregistrement (CE) correspond à l'effort fourni par les utilisateurs lors de l'enregistrement. Cela peut être, par exemple, le nombre de signatures à fournir dans le cas d'une reconnaissance par signatures manuscrites. Dans le cas de la dynamique de frappe, cela peut être la longueur de la séquence à taper par l'utilisateur.
- Le coût de reconnaissance (CR) correspond à l'effort fourni par les utilisateurs lors de la phase de reconnaissance. Pour la dynamique de frappe, on utilisera, par exemple, la longueur de la séquence à taper par un utilisateur lors de la phase d'authentification.

Il existe un autre critère important pour la mesure des performances d'un système biométrique. Ce critère correspond aux problèmes liés à la phase d'enregistrement et est défini par le taux d'échec à l'enregistrement (TEENR). Ce problème concerne des personnes qui ne réussissent pas à créer un profil. Cela peut être dû à des malformations physiques ou des évolutions et/ou à des variations rapides dans leur profil biométrique. Pour les empreintes digitales cela peut, par exemple, être le cas d'une personne manipulant des substances abrasives qui lissent les empreintes. Dans le cadre de la dynamique de frappe, cela peut correspondre à des personnes trop hésitantes, stressées lors de l'enregistrement ou en cours d'apprentissage de la frappe au clavier. Ces personnes fournissent bien un profil de dynamique de frappe mais celui-ci n'est pas exploitable et empêche toute reconnaissance correcte ultérieure de la personne. Ce taux est hélas difficilement quantifiable dans des conditions expérimentales. Il faut cependant prévoir des

solutions pour ce genre de personnes, même si ces problèmes n'apparaissent pas forcément lors de la phase de conception.

Une fois les indicateurs de performance choisis, il reste à les évaluer. En effet, il n'est jamais possible d'avoir connaissance de la valeur des performances sur la totalité de la population visée : l'objectif est d'obtenir des estimations les plus fiables possibles de ces indicateurs. Pour ce faire, ils seront évalués sur une base de test. Quand cela est possible et afin de pouvoir comparer les résultats d'une méthode à l'autre, l'utilisation de base publique commune à tous les chercheurs doit être systématique. Un grand nombre de bases couvrant diverses méthodes biométriques existent (pour les signatures, pour l'iris, pour la voix ….). Mais pour quelques méthodes biométriques récentes ou peu utilisées, il n'existe pas encore de base de test publique. C'est le cas pour la dynamique de frappe qui nous intéressera dans la suite. Dans ce cas, la comparaison des résultats n'est pas facile et doit être réalisée avec prudence.

Le protocole de test doit aussi être clairement défini. Un bon protocole de test doit séparer les données en trois bases distinctes :

- La base d'apprentissage qui contient les données servant à créer les classificateurs
- La base de validation qui permet d'évaluer les paramètres des classificateurs
- La base de test sur laquelle les indicateurs de performances sont calculés

Quand les données sont présentes en très grand nombre, la création de ces trois bases ne pose pas de problèmes majeurs. Par contre, quand les données sont limitées en nombre, la division de la base ne peut parfois pas se faire aussi nettement, pour cela un certain nombre de stratégies de partitionnement existe afin de garder une bonne estimation des performances [Eriksson *et al.*, 2000], [Stone, 1977] et [Martens et Dardenne, 1998]. Citons notamment :

- la validation croisée
- le *Leave One out*
- la séparation en groupes aléatoires

Ces stratégies permettent d'obtenir des indicateurs de bonne qualité, même avec des bases de tailles assez faibles.

1.2. L'authentification biométrique : un problème à une classe

Nous avons vu que l'authentification biométrique peut être résolue de deux façons différentes. La première d'entre elles propose de mettre en place un système de classification classique (identification). Mais cette façon de faire ne peut pas être appliquée à toutes les situations réelles. En fait, cela n'est possible que si nous avons à notre disposition des données d'autres utilisateurs pour construire les classificateurs. Ceci est impossible dans beaucoup de systèmes biométriques pour diverses raisons (par exemple pour la dynamique de frappe quand le mot de passe est différent pour tous les utilisateurs). Dans la suite de nos travaux, nous traitons uniquement de la résolution des problèmes d'authentification en les considérants comme les problèmes de classification à une classe. Nous présentons donc dans cette partie, une définition de la classification à une classe et effectuons un bref état de l'art résumant les problématiques et solutions proposées dans cette thématique.

1.2.1. Problème de classification à une classe

1.2.1.1. En disposant de données d'imposteurs

Pour bien comprendre la spécificité de la classification à une classe nous commençons par évoquer le problème à deux classes. L'objectif de la classification à deux classes est de séparer deux types de données que nous étiquetons par OUI et NON. Ces données sont représentées par des vecteurs numériques. Pour résoudre le problème de classification à deux classes, on détermine une « frontière » (Figure 8) séparant l'espace en deux zones. Cette frontière peut prendre différentes formes (un hyper-plan, une hyper-sphère, un regroupement d'hyper-sphères….).

Pour construire cette frontière, les méthodes classiques s'appuient sur des individus/données connues a priori. L'ensemble de ces données est appelé ensemble

d'apprentissage. La construction de la frontière s'appelle l'apprentissage. Si les données sont étiquetées, c'est-à-dire si on connaît pour chacune sa classe, on réalise un apprentissage supervisé. Si les données ne sont pas préaffectées à une classe, on parle d'apprentissage non supervisé. L'objectif est alors de créer des classes dans l'ensemble d'apprentissage. A l'issue de l'apprentissage, nous disposons d'un classificateur. Ce classificateur détermine, pour un nouvel individu, sa position par rapport à la frontière et lui attribue une classe en conséquence.

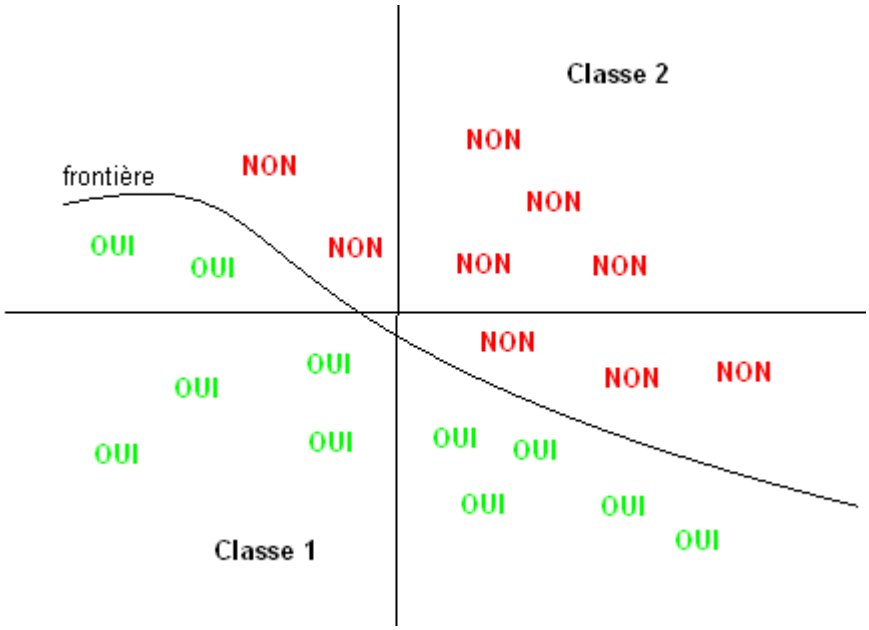


Figure 8 : Séparation de deux classes par un classificateur créant une frontière non linéaire

Pour créer le classificateur, de grandes quantités d'information sur les deux classes sont utilisées. La plus évidente d'entre elles est la position dans l'espace des données des deux classes.

Parmi les informations qui aident à la séparation des classes se trouvent également les centres de gravités des classes, ainsi que les variances intra et interclasse. Ces informations donnent une indication sur la facilité de séparation des classes.

La résolution du problème peut se faire aussi par des mesures statistiques ou de probabilités calculées sur les deux classes : la distribution des deux classes est alors modélisée par des lois de probabilité.

La présence de données étiquetées dans l'ensemble d'apprentissage permet d'utiliser des classificateurs très performants comme par exemple les réseaux de neurones ou les séparateurs à vaste marge (SVM).

Il est également possible avec deux classes connues de mettre en place des stratégies de sélection de caractéristiques et de détermination des paramètres du système.

1.2.1.2. Sans données d'imposteurs

Lorsque seules des informations sur une unique classe sont disponibles dans l'ensemble d'apprentissage, on parle de classificateurs à une classe. Le classificateur doit alors détecter les observations trop différentes de celles connues mais il ne possède, a priori, aucun contre exemple.

La Figure 9 présente un exemple d'ensemble de données appartenant à une classe à partir duquel il faudra être capable de décider si d'autres individus appartiennent à la même classe ou non.

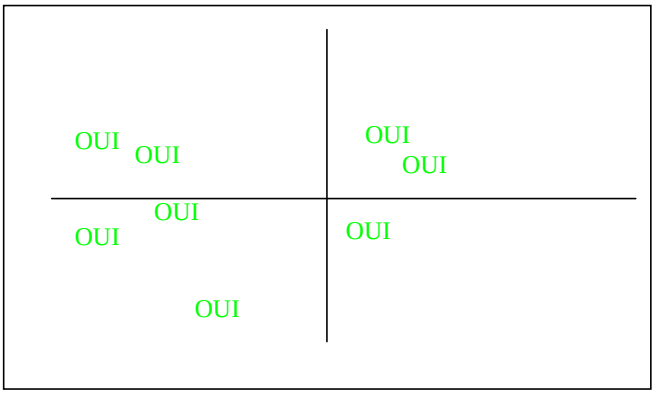


Figure 9 : Ensemble d'apprentissage d'un problème à une classe

L'objectif de la classification à une classe est de déterminer les zones de l'espace de représentation dans laquelle se trouvent les données de la classe connue. Il est parfois préférable et plus simple de déterminer plusieurs sous-divisions de l'espace de différentes formes et de différentes tailles de manière à englober les données de l'ensemble d'apprentissage. Il faut ensuite, pour une nouvelle donnée, décider si elle appartient à la classe ou non. Pour cela il ne faut pas que la division de l'espace soit trop resserrée autour des données connues rejetant ainsi les données de la classe qui ne sont pas encore connue. Inversement cette division de l'espace ne doit pas inclure trop d'espace autour des données connues sinon le risque d'englober trop de données n'appartenant pas à la classe devient trop important.

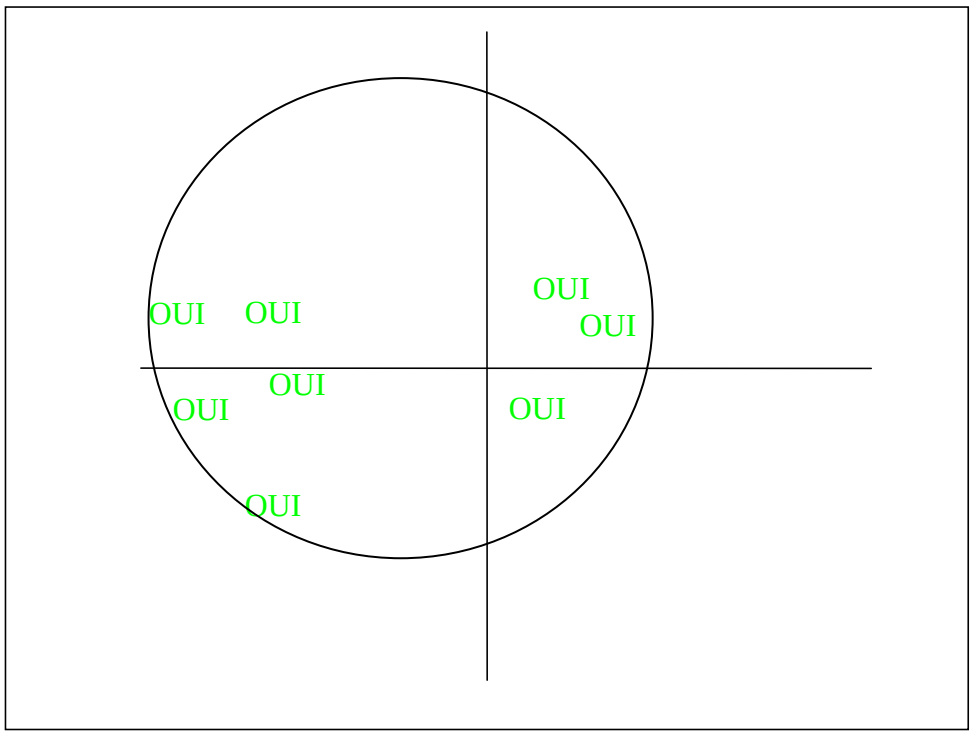


Figure 10 : Résolution d'un problème à une classe à partir de la position de l'ensemble des observations

Par exemple, sur la Figure 10, la frontière qui sépare la classe du reste de l'espace est définie par un cercle qui englobe toutes les données de l'ensemble d'apprentissage. Les caractéristiques de ce cercle ont été déterminées en considérant le segment entre les deux données les plus éloignées comme diamètre. Dans ce cas, toutes les données connues de la classe sont incluses dans la zone délimitée par le cercle et le risque de rejeter des éléments de cette classe est faible. Par contre, le risque de classer dans cette classe, un grand nombre de données qui ne devraient pas être considérées comme similaires est très important.

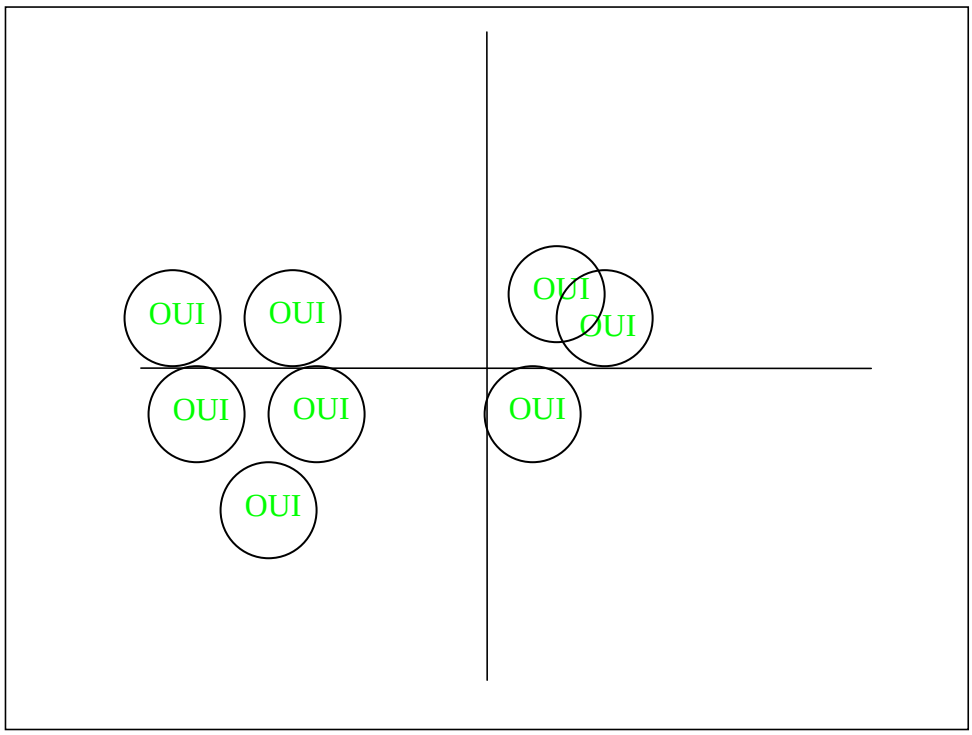


Figure 11 : Résolution d'un problème à une classe à partir de la position de chaque observation

La Figure 11 montre qu'il existe une autre possibilité pour caractériser la classe connue. Avec cette seconde approche la classe est vue comme la réunion de petits disques centrés autour de chaque donnée d'apprentissage et de rayon assez faible. Avec un petit rayon, cette séparation de l'espace minimise la chance d'accepter des donnes de façon incorrecte mais le risque d'écarter à tord beaucoup de données de la classe qui s'écartent un peu des données connues augmente alors proportionnellement.

La différence entre la Figure 10 et la Figure 11 provient du nombre de zones de l'espace que l'on décide d'utiliser pour caractériser la classe connue. La division de l'ensemble des données en classes est réalisée avec des algorithmes de classification non supervisée, ce nombre peut varier de 1 à n (avec n le nombre de représentants de la classe dans l'ensemble d'apprentissage cf. Figure 11). Dans ce cas, une hypothèse doit être utilisée pour choisir a priori le nombre de zones à utiliser. La justification de cette hypothèse peut se trouver dans des connaissances a priori sur la classe comme par exemple la façon dont les données sont générées. Elle ne pourra hélas être validée de façon définitive qu'a posteriori.

La classification à une classe comporte donc plusieurs difficultés :

- La première est de trouver le bon compromis entre limiter la zone de l'espace autour des données connues et étendre celle-ci pour couvrir un maximum d'espace. Ce problème qui est toujours présent dans l'authentification (compromis entre le TFR et le TFA) devient critique dans le problème à une classe du fait de l'absence d'information sur les données d'autres classes et du petit nombre de représentants connus (contenus dans le profil biométrique).
- La seconde difficulté est la détermination de la forme de la zone de l'espace contenant les données de la classe. Les structures qui apparaissent dans l'ensemble d'apprentissage ne sont pas obligatoirement caractéristiques de la classe. Elles peuvent tout aussi bien être caractéristiques de la manière dont ont été générées les données. Ainsi, les données n'appartenant pas à la classe auront la même structure : dans ce cas le risque est d'apprendre le phénomène plutôt que la classe ! Cette difficulté peut se rencontrer dans la dynamique de frappe : le risque est d'apprendre la séquence de touches tapée plutôt que le comportement propre des utilisateurs. Le petit nombre de données présentes dans le profil pose également une nouvelle fois problème car il est difficile de déterminer une forme à partir de 5 ou même 10 exemples.

Un autre point peut compliquer énormément la résolution du problème : Que se passe-t-il si l'ensemble d'apprentissage comporte un ou plusieurs éléments erronés ? Ces données peuvent être présentes à cause d'une défaillance d'un capteur, d'une fausse manipulation de l'utilisateur, d'une distraction de celui-ci, ce qui est courant dans le domaine de la biométrie comportementale. La présence de ces données incorrectes perturbe grandement la résolution en forçant l'acceptation d'une plus grande partie de l'espace.

Ces difficultés rencontrées lors de la résolution d'un problème à une classe se retrouvent dans tous les systèmes d'authentification biométrique et influent de manière importante dans la composition de leur architecture.

1.2.2. La phase d'extraction de caractéristiques

La première étape d'un problème d'authentification biométrique est de passer des caractéristiques physiques d'un individu à des caractéristiques informatiques. Il s'agit de transformer les données brutes fournies par les capteurs lors de l'acquisition des données en vecteur de données (numériques ou non) pouvant être présenté à un classificateur pour obtenir une décision.

L'extraction des caractéristiques est propre aux problèmes étudiés (extractions des minuties pour les empreintes digitales, des temps entre deux touches pour la dynamique de frappe…). Cette phase pose néanmoins des questions communes :

- La première question est comment choisir les caractéristiques qui seront réellement utilisables lors de la classification ? En effet, parmi toutes les caractéristiques extraites certaines n'apportent pas d'information pour aider à la résolution du problème et d'autres perturbent même le système, il faudra donc écarter certaines caractéristiques.
- La seconde question porte sur le choix, parmi toutes les données de l'ensemble d'apprentissage, de celles qui seront utilisées pour créer le profil de l'utilisateur. L'objectif est d'écarter les données bruitées du profil et de ne pas obtenir après l'enregistrement un profil du bruit plutôt qu'un profil de l'utilisateur.
- Une autre question se rajoute encore : doit-on utiliser les mêmes caractéristiques pour définir l'espace de représentation de tous les utilisateurs ? En biométrie, il est fréquent, que les caractéristiques qui peuvent être extraites pour un utilisateur ne puissent être sur d'autre. Cela arrive, par exemple, dans la reconnaissance de la voix quand les séquences prononcées par les utilisateurs pour être authentifiés sont propres à chacun. Les phonèmes ne sont donc pas les mêmes en type et en nombre, les caractéristiques extraites varient donc d'un utilisateur à l'autre. Cela peut aussi être un désir des concepteurs d'adapter les caractéristiques à chaque usager.

1.2.2.1. Sélection des échantillons constituant le profil d'un utilisateur

La phase d'enregistrement consiste en l'acquisition de plusieurs exemplaires d'une donnée biométrique afin de construire le profil d'un individu. Par exemple, il est demandé plusieurs échantillons de signatures manuscrites afin de construire un profil qui tiendra compte des particularités et des variabilités. Il arrive fréquemment que l'acquisition d'une donnée soit perturbée. Il est donc nécessaire de filtrer les données acquises afin d'éliminer celles qui ne sont pas représentatives de l'utilisateur.

Pour réaliser ce filtrage, des auteurs utilisent des méthodes basées sur des algorithmes de regroupement et de classification avec apprentissage non supervisé.

Dans [Uludag *et al.*, 2004], les auteurs évaluent deux méthodes de sélection. La première consiste à calculer pour chacun des exemplaires de la donnée biométrique, la distance moyenne avec toutes les autres. Les auteurs choisissent de garder les *k* données les plus similaires. Le risque est alors, si *k* est trop petit de perdre toutes les indications représentant la variabilité du profil.

La seconde méthode que les auteurs utilisent regroupe les données en cluster pour ensuite produire un représentant pour chaque cluster. Le regroupement des données est réalisé à partir du dendrogramme associé. On coupe le dendrogramme afin d'obtenir k groupes. Avec cette méthode, le risque est de tenir compte de données anormales se trouvant dans l'ensemble d'apprentissage.

Chacune de ces méthodes organise les observations issues de la phase d'enregistrement en groupes d'observations similaires et rejette toutes celles correspondant à des groupes de trop petites tailles.

Une approche par classification hiérarchique a été également utilisée dans [Kacholia et Pandit, 2003] avec succès.

Quand beaucoup de données sont présentes dans l'ensemble d'apprentissage, la détection des données incorrectes est, en elle-même, un problème à une classe. Ce type de méthodes, communément regroupées sous la dénomination « détection d'étranger », s'apparente à un problème à une classe. Quand trop peu de données sont disponibles, il est difficile de faire plus qu'éliminer les données complètement aberrantes car le risque de perdre une grande partie de l'information sur un utilisateur devient trop important.

1.2.2.2. Réduction du nombre des caractéristiques

Comme nous l'avons vu, l'utilisation d'une sélection de caractéristiques est souvent indispensable pour écarter les caractéristiques non pertinentes ou redondantes. Un autre point plaide en faveur d'une sélection des caractéristiques : la plupart des classificateurs fonctionnent mieux dans des espaces de faible dimension comme les réseaux de neurones par exemple. Ceci est dû au phénomène décrit par Bellman [Bellman, 1961] : *« the curse of dimensionality »* qui a montré que pour un même nombre de données d'apprentissage, l'augmentation de la dimension de l'espace de représentation de celles-ci empêche de déterminer une structure dans l'espace. Les données sont alors isolées dans de grands espaces de vide. Dans la biométrie, le nombre de données d'apprentissage étant souvent limité, il est donc nécessaire de procéder à une réduction du nombre de caractéristiques.

Cette réduction peut être effectuée de différentes façons :

- Sélectionner des caractéristiques pour ne garder que les meilleures
- Créer de nouvelles caractéristiques en en résumant plusieurs

Les caractéristiques choisies peuvent être communes à tous les utilisateurs, ou propres à chacun. Pour sélectionner les caractéristiques les plus discriminantes, la méthode la plus simple est d'utiliser des connaissances a priori. Par exemple, dans [Monrose et Rubin, 2000], les auteurs, qui travaillent sur la dynamique de frappe, se limitent à étudier les caractéristiques extraites des couples de touches les plus présentes dans l'anglais écrit en les supposant plus discriminants car présents en plus grand nombre dans les textes et donc dans la séquence de reconnaissance.

Le moyen le plus efficace pour sélectionner les caractéristiques est d'explorer l'ensemble des regroupements de caractéristiques possibles puis de les évaluer à l'aide d'une fonction de performance. Le problème est de construire cette fonction. Dans le problème de classification à plusieurs classes, cette fonction de performance est souvent construite à partir des taux de réussite du système évalués sur l'ensemble d'apprentissage.

Dans le problème à une classe, plusieurs cas se présentent. Si les utilisateurs ont tous les mêmes caractéristiques et s'il est possible d'obtenir des données de quelques imposteurs, une fonction de performance peut être déterminée même si l'échantillon est souvent de taille limitée.

Dans le cas général, la détermination de la fonction de performance est plus difficile. Sa construction ne peut pas s'appuyer sur les performances du classificateur lui-même. Il faut alors se baser sur d'autres éléments parfois difficiles à déterminer. Il s'agit néanmoins ici d'une étape indispensable.

Une fois la fonction de performance construite, plusieurs heuristiques existent pour explorer l'espace des caractéristiques possibles afin d'éviter une recherche exhaustive :

- L'algorithme Sequential Floating Forward Search algorithm (SFFS) [P.Pudil *et al.*, 1994] peut être utiliser pour sélectionner les caractéristiques. SFFS permet souvent l'obtention d'un bon ensemble de caractéristiques dans un laps de temps raisonnable.
- Une autre solution consiste à utiliser des algorithmes génétiques (AG) [Mitchell, 1996]. Les AG visent à mimer le processus de l'évolution d'une population. Les AG permettent souvent d'obtenir de très bonnes performances et d'avoir une bonne exploration de l'espace des solutions. Leur limitation est par contre liée au temps nécessaire pour obtenir une bonne solution.

Une autre façon de réduire le nombre de caractéristiques est d'en créer des nouvelles plus pertinentes par combinaisons (linéaires ou non) des caractéristiques initiales. Les méthodes les plus populaires et puissantes utilisées pour réaliser ces combinaisons sont les analyses en composantes. Par exemple, l'analyse en composantes principales (ACP) [Benzécri, 1973] transforme l'espace des caractéristiques en un nouvel espace. Les axes du nouvel espace déterminent des combinaisons linéaires des caractéristiques. A chaque axe du nouvel espace est également associé un coefficient qui représente le pourcentage de l'inertie du nuage initial expliqué par l'axe. Les nouvelles caractéristiques sélectionnées seront ensuite les axes ayant les plus forts taux d'explication de l'inertie. L'inconvénient de cette solution est que rien n'assure que ces caractéristiques sont les meilleures pour discriminer les utilisateurs, même si les axes de faible inertie sont souvent considérés comme des axes expliquant le bruit

La sélection de caractéristiques reste un problème délicat dans le cadre de la classification à plusieurs classes. Elle ne peut être appliquée au problème à une classe que lorsque tous les utilisateurs ont les mêmes caractéristiques et lorsque l'on dispose pour certains utilisateurs d'attaques d'individus n'appartenant pas à la classe connue. Les caractéristiques sélectionnées pour ces utilisateurs seront ensuite

utilisées pour tous. Dans le cas où les utilisateurs ont des caractéristiques différentes, aucune solution n'est aujourd'hui satisfaisante du fait de la difficulté d'évaluer la performance des caractéristiques.

1.2.2.3. Normalisation des caractéristiques

Le rôle de la normalisation est, d'une part d'éviter les influences des facteurs d'échelle quand les données varient dans des intervalles différents. D'autre part, la normalisation permet de gommer les effets des rapports moyenne/variance pour éviter que l'une des caractéristiques cause à elle seule le rejet systématique d'un utilisateur.

Il existe un grand nombre d'opérateurs de normalisations classiquement utilisés et présentés notamment dans [Jain *et al.*, 2005]. Pour normaliser une donnée, il est nécessaire de posséder des informations sur sa distribution. Les informations utilisables sont le maximum max_X, le minimum min_X, la moyenne μ_X, l'écart type σ_X de la variable à partir desquelles une transformation mathématique est appliquée à la donnée X afin d'obtenir la donnée normalisée X_{norm}.

$$X_{norm} = \frac{X}{\max_X} \qquad \textbf{(1)}$$

$$X_{norm} = \frac{X - \min_X}{\max_X - \min_X} \qquad \textbf{(2)}$$

$$X_{norm} = \frac{X - \mu_X}{\sigma_X} \qquad \textbf{(3)}$$

Les équations (1) (2) et (3) présentent trois opérateurs qui peuvent être utilisés pour la normalisation. Les opérateurs (1), (2) sont les moins performants mais ont l'avantage de nécessiter des données plus faciles à estimer. L'opérateur (3) est le plus performant mais implique d'estimer la moyenne et la variance, ce qui peut être difficile. Dans ce dernier cas, on suppose la distribution normale.

En sortie de la phase d'extraction des caractéristiques, le système dispose de vecteurs de caractéristiques numériques pour chaque utilisateur que nous appelons observations par la suite.

1.2.3. Méthodes de classification à une classe

Pour résoudre un problème de classification à une classe, certains auteurs [Tax *et al.*, 1999; Tax et Duin, 2002; Tax et Duin, 2004] proposent d'adapter des classificateurs multi-classes plutôt que de créer de nouveaux classificateurs spécifiquement pour ce problème. Une bonne présentation des différents classificateurs à une classe est proposée dans [Markou et Singh, 2003] et [Markou et Singh, 2003]. Dans la partie qui suit, nous complétons ces articles avec des travaux plus récents et ajoutons nos commentaires sur l'intérêt des différentes méthodes proposées.

1.2.3.1. Mesures de similarité

Les mesures de similarité ou de dissimilarité basées sur des calculs de distances sont parmi les moyens les plus simples pour obtenir un score indiquant la proximité entre deux vecteurs de caractéristiques. Dans un problème à une classe, il suffit ensuite de comparer ce score avec un seuil afin d'obtenir la décision finale.

Les différentes possibilités existantes pour décider de la validité d'une observation par calcul de similarité sont :

- Comparaison avec le vecteur moyen des vecteurs de l'ensemble de l'apprentissage
- Calcul de la moyenne des similarités/dissimilarités calculée par rapport à tous les vecteurs de l'ensemble d'apprentissage
- Calcul des similarités/dissimilarités maximum et minimum par rapport à tous les vecteurs de l'ensemble d'apprentissage

Les mesures de dissimilarité ou similarité ont pour avantage d'être très simples à utiliser, d'avoir une complexité très faible et de ne nécessiter que peu de données dans l'ensemble d'apprentissage et de n'avoir aucun paramètre à fixer (à part le seuil de décision). Leur utilisation dans les problèmes à une classe est d'ailleurs identique à leur utilisation dans le cadre du problème à N classes. Malheureusement, leurs performances sont souvent assez faibles car ces mesures sont très sensibles aux bruits. De plus, leurs performances se détériorent rapidement dans des espaces de dimension élevée. Néanmoins, elles restent assez utilisées notamment dans [Monrose et Rubin, 1997]. Pour certains problèmes, des mesures de similarité spécifiques peuvent être développées, par exemple le *Dynamic Time*

Warping(DTW) [Martens et Claesen, 1997] est très souvent utilisé en reconnaissance de signatures manuscrites.

Afin d'obtenir de meilleurs performances, ces mesures peuvent être incorporée dans des classificateurs plus évolués comme par exemple les k-plus proches voisins qui seront présentés dans la suite.

1.2.3.2. Méthodes statistiques

Une autre famille de méthodes est basée sur l'utilisation de données statistiques comme la moyenne et la variance, couplées à l'utilisation de tests ou de mesures statistiques.

Dans le cas du problème à une classe, une des moyennes correspond aux données acquises lors de l'enregistrement et l'autre provient de l'observation à tester.

Différentes mesures ou tests peuvent être utilisées :

- Le T-test [Gosset, 1908] de comparaison de deux moyennes permet de comparer si deux échantillons d'une population ont une même moyenne. Ce test nécessite que les variances des deux échantillons soient identiques ce qui est souvent difficile à prouver. De plus la qualité du test est également très fortement dégradée par la taille des deux échantillons. Ce test n'est pas utilisé dans les applications mais il peut être utilisé dans des travaux préliminaires pour savoir s'il est possible de différencier deux individus [Gaines *et al.*, 1980].

- Test de Wilcoxon-Mann-Whitney mis en place dans ses deux versions par Wilcoxon et Mann-Whitney [Mann et Whitney, 1947],[Wilcoxon, 1945] permet de vérifier que deux échantillons d'une population suivent une même distribution. Il est un peu plus fiable que le T-test, mais avec les mêmes inconvénients.

Il existe de nombreux autres tests statistiques dont les conditions d'application et les hypothèses varient. L'avantage de ces tests est de permettre de décider (facilement et avec un faible coût de calcul) si une observation est émise par le même utilisateur que le profil, ces tests ne nécessitent que peu de données d'apprentissage. Néanmoins, on ne peut pas être sûr que l'hypothèse testée est celle qui permet de distinguer les imposteurs.

En dehors des tests statistiques, il existe des mesures statistiques permettant de tester si un vecteur de données est similaire à un ensemble de vecteurs connus. Le z-score en est un exemple. Le z-score est un score qui permet d'indiquer si une observation numérique x est ou non « loin » d'une population dont on connaît la moyenne μ et l'écart type σ. Son application au problème à une classe est donc directe, la population étudiée étant l'ensemble d'apprentissage associé à la classe. Il est calculé comme indiqué sur l'équation (4)

$$z = \frac{|x - \mu|}{\sigma} \qquad \textbf{(4)}$$

Dans le cas d'une loi normale, les statisticiens ont fourni des tables indiquant le pourcentage d'individus authentiques rejetés pour une valeur du z-score. Pour adapter le z-score à un espace à n dimensions, il existe plusieurs solutions : l'une d'entre elles est de compter le nombre de valeurs non valides pour un vecteur de caractéristiques. Le nombre de caractéristiques ne respectant pas le z-score, peut alors être comparé à un seuil. Pour cela, il est nécessaire d'avoir déterminé au préalable un vecteur des moyennes $\mu\{\mu_1, \mu_2, \ldots, \mu_n\}$ et un vecteur des écarts types $\sigma\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ pour les caractéristiques contenues dans le vecteur. Ces valeurs peuvent être calculées sur les vecteurs de données de l'ensemble l'apprentissage (profil de l'utilisateur). Le taux de caractéristique valide du vecteur $X\{X_1, X_2, \ldots X_n\}$ est calculé à partir de l'équation (5) qui nécessite de fixer un paramètre α.

$$|X_i - \mu_i| < \alpha\sigma_i \qquad \textbf{(5)}$$

Cette méthode revient donc à définir une sorte de tunnel autour des valeurs moyennes associées aux différentes caractéristiques disponibles dont la largeur est définie à la fois par l'écart type et par le paramètre α.

Un vecteur est validé si le taux de caractéristique non valide est inférieur à un seuil. Le z-score est facile à mettre en place pour un problème à une classe, mais pour qu'il soit efficace, il faut que les caractéristiques observées suivent une loi normale. Il faut aussi disposer d'un ensemble d'apprentissage suffisant pour estimer la variance et la moyenne (une dizaine est suffisante dans la plupart des cas). Il est souvent utilisé pour la dynamique de frappe [Leggett et Williams, 1988].

1.2.3.3. Chaînes de Markov Cachées (CMC)

Les Chaînes de Markov Cachées (CMC) [Rabiner, 1989] constituent un outil d'analyse stochastique puissant. Une des grandes difficultés des CMC consiste en la détermination de la modélisation correcte du phénomène, Il s'agit de définir à quoi correspondent les états cachés, quelles sont les observations émises par ceux-ci visibles de l'extérieur… Les CMC sont utilisées en biométrie ([Chen et Chang, 2004], [Ferrer *et al.*, 2005]). Une fois entraînées les CMC fournissent la probabilité qu'une observation est été émise par le modèle appris sur le profil de l'utilisateur. Cet outil est donc directement adapté aux problèmes à une classe. Ses principaux désavantages sont le nombre d'observations important nécessaire pour l'apprentissage (les auteurs utilisent une cinquantaine d'observations pour apprendre le modèle) et la création d'un modèle correct du phénomène. Mais si l'adaptation est bien réalisée, les CMC sont alors très performantes.

1.2.3.4. Estimation de densité

Une des manières d'identifier la zone de l'espace de dimension *n* dans laquelle se trouvent les données d'une classe, est d'estimer pour chaque point la probabilité qu'une donnée appartienne à la classe étudiée. Les méthodes présentées dans cette partie sont basées sur l'estimation de densité. Toutes ces méthodes nécessitent donc uniquement des données permettant d'estimer les densités de probabilités, elles sont donc directement adaptées aux problèmes à une classe.

- Estimation par une loi gaussienne

La distribution la plus courante pour modéliser des distributions de données est la distribution normale ou gaussienne. Pour calculer les densités de probabilité, le vecteur moyen μ et la matrice de covariance $\sum$ doivent être stockés dans le profil de l'utilisateur. Ces valeurs doivent être calculées sur les données d'apprentissage qui doivent donc être en nombre suffisant. L'inversion de la matrice de covariance est très coûteuse en temps de calcul ce qui explique que l'on utilise souvent une approximation $\sum^{+}$ (équation (6)).

$$\Sigma^{+} = \Sigma^{T}\left(\Sigma\Sigma^{T}\right)^{-1} \qquad \textbf{(6)}$$

Le problème de ce modèle est l'hypothèse faite sur les données : ces dernières doivent correspondre à une unique loi gaussienne. C'est pourquoi plutôt que de

n'utiliser qu'une seule loi gaussienne, on modélise souvent une distribution par une mixture de gaussiennes.

- Mixture de gaussiennes [McLachlan et Peel, 2000]

Les mixtures de gaussiennes sont l'estimation de la densité de probabilité à l'aide de plusieurs gaussiennes. Les performances sont souvent meilleures que quand on utilise une unique gaussienne.

Le problème de la mixture de gaussiennes, est que plus on augmente le nombre de gaussiennes, plus on a de paramètres à estimer (les centres et les matrices de covariance). Se pose alors la question de la validité du modèle lorsque peu de données sont disponibles pour l'apprentissage. De plus le nombre de gaussiennes entraîne également une forte hausse de la complexité du calcul. Néanmoins les mixtures de gaussiennes sont beaucoup utilisées en biométrie [Richiardi et Drygajlo, 2003] et [Stylianou *et al.*, 2005].

- Les fenêtres de Parzen

Les fenêtres de Parzen [Parzen, 1962] consistent à pousser la logique précédente à l'extrême en plaçant une gaussienne en chaque point de l'ensemble d'apprentissage.

Le grand avantage de la méthode des fenêtres de Parzen est l'absence d'apprentissage. Ses inconvénients sont le stockage de tous les points de l'ensemble d'apprentissage et le test d'un nouveau point qui est assez long. En effet, chaque point de l'apprentissage doit être examiné. Un autre désavantage majeur est la performance de la méthode qui dépend énormément de l'échantillonnage de l'ensemble d'apprentissage. Si peu de points sont disponibles ou le sont de façon mal répartie pour une classe, de très mauvaises performances sont à prévoir. Les fenêtres de Parzen ont été utilisées dans des systèmes de détection d'intrusions réseaux [Yeung et Chow, 2002] et dans différents systèmes biométriques [Prabhakar et Jain, 2002] .

Nous avons vu comment estimer la densité de la distribution des observations d'une classe en tous points de l'espace. Nous allons maintenant étudier d'autres méthodes permettant d'identifier la zone de l'espace dans laquelle se trouvent les observations d'une classe, ces méthodes cherchent à délimiter par une frontière la zone de l'espace où se trouvent les éléments de la classe. Cette frontière peut être définie par des points, des hyperplans ou bien des hyper-sphères.

1.2.3.5. k-plus proches voisins

La méthode des k-plus proches voisins [Belur, 1991] est probablement la plus simple pour résoudre des problèmes de classification. Pour un nouveau vecteur à tester, il faut identifier les *k* vecteurs les plus proches dans l'ensemble d'apprentissage. La nouvelle observation est attribuée à la classe la plus représentée parmi ces *k*. Dans le cas du problème à une classe des stratégies un peu différentes doivent être utilisées du fait de l'existence d'une seule classe.

La première adaptation possible de cette méthode est de travailler avec les plus proches voisins en ne regardant pas la classe la plus présente mais en comparant la moyenne des *k* distances à un seuil de décision.

Une autre façon de faire est de comparer le rapport *R* entre la distance de l'observation *X* à classer avec son plus proche voisin Y_j dans l'ensemble d'apprentissage, et la distance entre Y_j et son plus proche voisin. Pour cela l'équation (7) est utilisée. $\Omega=\{Y_1,Y_2, \ldots,Y_i\ldots\}$ est l'ensemble d'apprentissage, *d(X,Y)* une mesure de distance, et *X* l'observation à tester. Le rapport est comparé à un seuil afin de prendre la décision sur *X*.

$$j = \arg\min_i (d(X, Y_i))$$
$$R(X) = \frac{d(X, Y_j)}{Min_i (d(Y_i, Y_j))} \qquad \textbf{(7)}$$

Dans [Tax, 2001], l'auteur propose une nouvelle méthode basée sur une comparaison de densités. Dans cette méthode, il fait grandir une sphère de rayon *R* autour du point testé jusqu'à ce qu'elle englobe les *k* plus proches voisins de ce point. Ensuite, il calcule une densité locale dans la sphère avec l'équation (8). *X* est l'observation testée, *Nb* le nombre d'objets dans la base d'apprentissage, PVP_k le *kième* plus proche voisin de *X* et *Vk* le volume de l'hypersphère qui ne dépend que de la distance au *kième* plus proche voisin. L'auteur rejette ensuite *X* si cette densité est inférieure à celle calculé en prenant comme centre le plus proche voisin de *X*.

$$p_{NN} = \frac{k / Nb}{V_k(\left\| X - PPV_k(X) \right\|} \qquad \textbf{(8)}$$

L'avantage des méthodes basées sur les k-plus proches voisins est qu'elles ne nécessitent pas d'apprentissage, et qu'elles permettent une bonne prise en compte des

variations au sein du profil d'un utilisateur. Il est d'ailleurs très facile de réutiliser les mesures de distances de tous types. Leur grand inconvénient est que le test d'une nouvelle observation est assez coûteux en temps puisque il faut examiner la totalité des points de l'ensemble d'apprentissage, même si ce problème est moins présent dans le cadre de la biométrie où les ensembles d'apprentissage sont limités. Un autre inconvénient de ces méthodes est leur difficulté à traiter les espaces de grande dimension.

1.2.3.6. Séparateurs à vastes marges (SVM)

Les SVM sont des outils très utilisés dans le domaine de la classification depuis qu'ils ont été présentés par Vapnik [Vapnik, 1995]. Avant de détailler les méthodes d'adaptation des SVM au problème à une classe, il est nécessaire de faire un bref rappel sur leur utilisation classique. Leur objectif premier est de séparer deux classes. Pour réaliser cet objectif, l'espace est séparé en deux par un hyperplan défini par un vecteur normal *w*, et une constante *b* suivant l'équation (9).

$$w.X + b = 0 \qquad \textbf{(9)}$$

Cet hyperplan a pour propriété de maximiser une marge définie comme la distance minimale entre l'hyperplan et les points des deux classes.

Les SVM sont performants sur les problèmes classiques et ne nécessitent que peu de paramètres à fixer. Leur adaptation au problème à une classe est par contre plus problématique puisque l'ensemble d'apprentissage ne comporte des individus que d'une classe.

Schölkopf est le premier à avoir proposé une solution pour résoudre le problème à une classe à l'aide des SVM [Schölkopf *et al.*, 2001; Schölkopf *et al.*, 2000].

Il considère l'origine du repère comme unique représentant de la deuxième classe. Le principe est, d'isoler dans l'espace la zone contenant les *n* observations connues à l'aide d'un noyau adapté (souvent un noyau gaussien).

$$\begin{cases} \min \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(X_i . X_j) \\ \sum_{i=1}^{n} \alpha_i = 1 \end{cases} \quad \textbf{(10)}$$

Dans ce cas, le système à résoudre est alors indiqué sur l'équation (10). $\{X_1, X_2 \ldots X_n\}$ est l'ensemble des n vecteurs d'observations de l'ensemble d'apprentissage et les paramètres à optimiser sont $\{\alpha_1, \alpha_2, \ldots \alpha_n\}$.

L'avantage d'utiliser les SVM à une classe réside dans la possibilité de représenter une classe par des frontières non linéaires et ce de façon assez performante. Par contre la difficulté des SVM à une classe est de nécessiter un bon rapport [dimension de l'espace]/[taille de l'ensemble d'apprentissage] afin d'obtenir une bonne capacité de généralisation.

Une autre solution pour adapter les SVM à la classification à une classe a été proposée dans [Manevitz et Yousef, 2002]. L'approche est basée sur l'hypothèse que tout ensemble d'apprentissage de grande taille contient des données qui pourront être considérées comme étrangères à la classe à isoler. Ces données peuvent être présentes par exemple à cause de bruit ou d'une fausse manipulation de l'utilisateur.

Nous avons déjà mentionné que la détection de ces données étrangères est un problème en soit. Cette détection est le point faible de la méthode, pour un problème biométrique car il n'y a aucune certitude que l'ensemble d'apprentissage contienne des données incorrectes.

Ainsi, cette méthode semble difficile à utiliser à part dans des cas très particuliers (et probablement pas dans des problèmes biométriques).

- Support Vector Data Domain (SVDD)

Dans [Tax, 2001; Tax et Duin, 1999], il est proposé une approche novatrice pour résoudre le problème à une classe basée sur une extension des SVM. Plutôt que d'essayer de séparer la classe à identifier du reste de l'espace par un hyperplan dans les espaces du noyau, l'auteur utilise une hyper-sphère pour englober la classe connue (Figure 12).

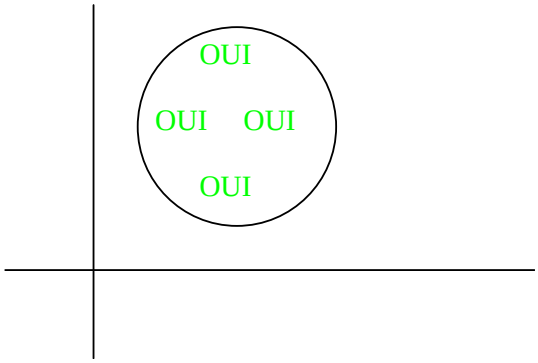


Figure 12: Hyper-sphère englobante

Cette hyper-sphère est définie par son rayon *R* et son centre *A*. Dans les SVM, l'objectif est de maximiser la marge entre l'hyperplan séparateur et les observations les plus proches appelées vecteurs de support. Dans les SVDD, l'objectif est de trouver la sphère de rayon minimal permettant d'englober les exemples de la classe connue.

La fonction objective ε à minimiser est présentée équation (11) et les contraintes avec les contraintes associées équation (12).

$$\varepsilon(R, A) = R^2 \qquad \textbf{(11)}$$

$$\|x_i - A\|^2 \leq R^2 + C\sum_i \varepsilon_i \qquad \textbf{(12)}$$

C est un paramètre indiquant la tolérance aux erreurs, ε_i indique si le point i est ou non dans la sphère.

Comme dans le cas des SVM, on peut remplacer les produits scalaires par une fonction noyau. Il est également possible d'ajouter quelques observations fausses connues, pour améliorer les performances. Il semble, vu les résultats obtenus dans des études précédentes [Manevitz et Yousef, 2002] [Tax, 2001; Tax et Duin, 1999] que les SVDD partagent les mêmes avantages/inconvénients que les SVM à une classe mais en étant un peu plus performants dans les espaces de forte dimension.

Ces diverses adaptations des SVM aux problèmes à une classe montrent qu'ils obtiennent des performances intéressantes pour peu que l'on se trouve dans un espace de dimension faible et que l'on dispose d'un ensemble d'apprentissage important.

1.2.3.7. Réseaux de neurones

En règle général, les réseaux de neurones ont besoin d'observations issues de toutes les classes à séparer afin d'entraîner le réseau. Il existe néanmoins une possibilité pour adapter certains types de réseaux de neurones au problème à une classe. En effet, parmi les réseaux de neurones, certains ont pour objectif de réaliser une classification non supervisée. Ils organisent les données d'entrée en créant eux-mêmes les classes. Parmi ces méthodes, les cartes de Kohonen (Self Organizing Map) [Kohonen, 1989] semblent pouvoir être adaptées assez facilement au problème à une classe :

- Il est possible d'entraîner le réseau à partir des données de la classe connue. Ensuite, pour décider si une observation d'entrée est authentique ou non, la solution la plus simple consiste à tester si l'observation active un neurone qui avait été activé par une des données de l'ensemble d'apprentissage. Dans le cas contraire, l'observation est rejetée.
- La seconde solution consiste à mesurer l'activation du neurone et de prendre la décision en comparant cette mesure à un seuil.

Une évolution des cartes de Kohonen a donné naissance à la méthode « *Learning Vector Quantization* » (LVQ) [Kohonen, 1998]. Leur objectif est d'approximer la distribution des classes par des « *code books* » (prototype de la classe) qui définissent des régions de Voronoï en réduisant au maximum la distance entre chaque prototype et le *code book* qui lui correspond. Dans le cadre du 1-LVQ, l'entraînement est limité aux données provenant de la classe à reconnaître et a pour objectif de produire des seuils explicites pour l'acceptation ou le rejet d'une observation. Pour optimiser la méthode, il est possible de fixer un seuil par région de Voronoï.

L'Adaptative Resonance Theory (ART) est mise en place pour permettre à certains réseaux de neurones d'évoluer continuellement au cours du temps, et de s'adapter à l'apparition de nouvelles classes. Les réseaux de l'ART existent en plusieurs types avec des données d'entrée binaires (ART-1) ou bien avec des donnée d'entrée quelconques (ART-2 : [Carpenter et Grossberg, 1987]) et (ART-2a). La classe de sortie est indiquée par l'activation d'une sortie. Une nouvelle sortie est crée lorsque aucune sortie courante ne correspond. Son adaptation au problème à une

classe se fait simplement en rejetant les observations nécessitant la création d'une nouvelle sortie.

Il semble finalement que les méthodes basées sur les réseaux de neurones sont mal adaptées aux problèmes à une classe à part les 1-LVQ qui semblent pourvoir être utilisés. Tout au moins, les réseaux de neurones semblent difficilement adaptables aux problèmes qui nous intéressent, c'est-à-dire à la détection de données ne provenant pas de la même source que celle dont provient l'ensemble d'apprentissage. Ils ont plutôt été conçus pour répondre au problème de l'organisation des données en classes. Néanmoins, ces méthodes peuvent aider à détecter les données bruitées au sein de l'ensemble d'apprentissage.

1.2.3.8. Bilan

Le Tableau 1 présente un résumé des différents classificateurs applicables aux problèmes à une classe. Ce tableau montre qu'il n'y a pas de solution miracle ou de bon classificateur adapté à tous les domaines. En fonction des problèmes à traiter, il faut choisir soigneusement le ou les classificateurs répondant aux contraintes (dimension de l'espace des caractéristiques, taille de l'ensemble d'apprentissage, connaissance a priori sur les données). Si l'ensemble d'apprentissage est de petite taille par rapport à la dimension de l'espace, les méthodes statistiques sont les plus adaptées. Si on dispose d'un ensemble d'apprentissage important, des méthodes plus performantes comme les SVM à une classe ou les SVDD peuvent être utilisées. Enfin, si quelques observations d'imposteurs sont disponibles ou s'il est possible d'en générer, un classificateur à une classe permettant de prendre en compte des observations d'imposteurs afin de mieux ajuster ses performances (comme les SVM à une classe) permettra d'avoir d'excellents résultats. Des modélisations par les CMC ou par des distributions de probabilité peuvent également donner de bons résultats lorsque des informations sur la forme des distributions des données ou la façon dont elles sont générées sont disponibles. Dans tous les cas, il est également très important de tenir compte du nombre et de la sensibilité des paramètres associés aux classificateurs retenus. Nous présentons une discussion plus approfondie à ce sujet dans la deuxième partie de ce manuscrit.

Tableau 1 : Bilan des performances des différents classificateurs utilisables sur le problème à une classe dans le cadre de la biométrie

Méthode	Avantages	Inconvénients	Cas d'utilisation
Mesures de similarité	Simplicité	Performance faible	A éviter, sauf dans des cas spécifiques, ou associées à une méthode plus complexe (k-ppv)
Méthodes statistiques	Simplicité Performances bonnes si distribution connue	Problème si les données suivent des distributions complexes	A utiliser, dans les cas où l'ensemble d'apprentissage est de faible taille (une dizaine d'observations)
Estimation de densité	Bonnes performances pour estimer des distributions dans des zones de l'espace	Nécessite un grand ensemble d'apprentissage (+50 observations)	Ces méthodes travaillent sur de grands ensembles d'apprentissage, et semblent moins performantes que les SVM à une classe
k-plus proches voisins	Aucun apprentissage	Performance très liée à la qualité de l'ensemble d'apprentissage	Utile pour une première approximation
-SVM à une classe ou -SVDD	Permet de séparer une classe non linéairement séparable, avec de bonnes performances	Nécessite un grand ensemble d'apprentissage (+50 observations)	A utiliser dès que l'ensemble d'apprentissage est suffisant
CMC	Très performant si le modèle est bien déterminé	Nécessite un grand ensemble d'apprentissage (+50 observations)	A utiliser dans des cas particuliers, suivant la modélisation du problème.
Réseaux de Neurones	Permet des séparations non linéaires	Nécessite un grand ensemble d'apprentissage (+50 observations)	Les LVQ peuvent être testés mais en règle générale ils semblent que les SVM permettent d'obtenir des résultats similaires

1.2.4. Classification à une classe et fusion

1.2.4.1. Intérêt de la fusion

Pour résoudre un problème biométrique, nous avons vu qu'il était possible d'utiliser plusieurs classificateurs différents. Le concepteur du système peut donc, soit se limiter à un classificateur, soit en choisir plusieurs qui seront ensuite combinés. Cette possibilité a été étudiée par plusieurs auteurs dans le cas du problème à une classe [Tax et Duin, 2001], [Lai *et al.*, 2002] ou en biométrie [Jain *et al.*, 2004] et [Fierrez-Aguilar *et al.*, 2005]. Nous examinons ici l'intérêt de choisir plusieurs classificateurs.

Quel que soit le domaine, avant de faire un choix, un bon décideur consulte plusieurs experts et combine leurs avis pour prendre la décision. Il nous parait donc intéressant d'étudier cette possibilité dans le cadre de l'authentification biométrique, c'est-à-dire dans le cadre de la classification à une classe.

Lors de la mise en place d'une étape de fusion le choix des classificateurs, est essentiel. Il n'est, par exemple, pas raisonnable de demander des avis à une série d'experts, dont nous savons qu'ils répondront tous de la même façon. De la même manière, en reconnaissance des formes, il est nécessaire de trouver des classificateurs présentant des points de vue différents sur les données c'est-à-dire ne se trompant pas sur les mêmes points de l'espace.

Le fait que deux classificateurs présentent des points de vue différents sur le système peut provenir de plusieurs facteurs :

- Ils utilisent des caractéristiques biométriques différentes (empreintes digitales et iris par exemple),.
- Ils utilisent différentes informations extraites d'une même caractéristique biométrique (image d'une signature manuscrite et dynamique de cette même signature).
- Ils fonctionnent de façons différentes, ce cas est plus difficile à identifier. Il se rencontre quand les classificateurs sont de natures différentes (réseau de neurones et k-ppv par exemple).

Dans [Dietterich, 2000], l'auteur donne plusieurs justifications au fait de combiner plusieurs classificateur:

- La première raison est d'ordre statistique. Chaque classificateur est construit pour minimiser une erreur sur un ensemble d'apprentissage. Il est possible de créer plusieurs classificateurs différents atteignant la même erreur sur l'apprentissage. Du fait de la taille des ensembles d'apprentissage souvent limitée, ces classificateurs auront des performances en généralisation différents. Utiliser plusieurs classificateurs réduits le risque d'avoir de mauvais très mauvaises performances en moyennant leur erreur.
- La seconde raison est liée à l'apprentissage : les classificateurs sont souvent entraînés à l'aide d'algorithmes d'optimisation numérique, et peuvent donc s'arrêter dans des minima locaux. L'utilisation de plusieurs classificateurs permet alors d'augmenter les chances d'approcher la solution optimale sur l'ensemble d'apprentissage.
- La dernière raison est liée à la qualité de la représentation des données par chaque classificateur : Même si, en théorie, certains classificateurs sont qualifiés d'approximateurs universels, les contraintes du problème (nombre de données disponibles, qualité…) font qu'un seul classificateur ne peut pas parvenir à lui seul à la séparation optimale d'une classe du reste de l'espace. La cause est, une fois encore la limitation de la taille de l'ensemble d'apprentissage. Le classificateur ne peut travailler que sur un nombre fini de divisions de l'espace. Plusieurs classificateurs permettent d'étendre le nombre de divisions de l'espace explorées.

1.2.4.2. Approche séquentielle (cascading)

L'approche séquentielle ou *cascading* [Alpaydin *et al.*, 2000] est une approche dont l'objectif est de soumettre le problème séquentiellement à plusieurs classificateurs de complexité croissante. L'avantage de l'approche séquentielle est d'accélérer le traitement dans le cas où la donnée permet une décision rapidement. En effet, la décision peut être obtenue sans avoir à passer par le second classificateur qui peut être plus gourmand en temps de calcul. Par contre cette approche n'est pas la plus performante. Son application à l'authentification biométrique ne se justifie que si le classificateur le plus performant est très coûteux en temps, et que le temps est une contrainte importante du système, ce qui est assez rare.

1.2.4.3. Méthodes basées sur les votes

Une autre façon de prendre en compte plusieurs classificateurs est de les faire voter. Dans ce cas, les classificateurs ne vont pas fournir un score mais une décision, c'est-à-dire la classe qu'ils donnent à l'observation à classer. Une fois les décisions des classificateurs connues, la décision finale est prise par un vote. Kittler dans [Kittler *et al.*, 1998] présente quelques unes des règles usuelles pour les votes. Les méthodes de fusion par votes permettent d'augmenter considérablement les performances du système.

1.2.4.4. Fusion des scores des classificateurs

Si les classificateurs fournissent des scores en sortie, il est possible de réaliser une combinaison des scores. La fusion peut avoir lieu avec l'une des règles de fusion définie par Kittler (*Somme*, *produit*…).

Les méthodes par combinaison des scores permettent de prendre en compte, non seulement la décision de chaque classificateur mais aussi son degré de confiance en permettant de fixer des poids à chaque classificateur.

Si les sorties des classificateurs sont sujettes au bruit ou peu stables ce qui est très fréquent en biométrie notamment comportementale, l'utilisation de la somme donne de meilleurs résultats alors que le produit peut dégrader très nettement les performances du système. Pour tenter de résoudre ce problème, on peut éliminer les valeurs des scores extrêmes (MAX, MIN).

Si les classificateurs sont indépendants, le produit donne les meilleurs résultats. Cependant, l'indépendance des classificateurs est une hypothèse rarement démontrable ou même acceptable si on travaille sur une unique caractéristique biométriques.

C'est pourquoi nous pensons que l'opérateur Somme est le mieux adapté pour l'authentification biométrique.

1.2.4.5. Fusion par des classificateurs

Les scores fournis par les différents classificateurs peuvent être également regroupés au sein d'un vecteur de caractéristiques qui sera présenté à classificateur un afin d'obtenir la décision finale. Ce classificateur devra avoir été entraîné auparavant à l'aide d'une base d'apprentissage contenant des vecteurs de scores et la décision associée.

Le classificateur peut être un réseau de neurones [Wang *et al.*, 2003] ou bien un SVM [Ben-Yacoub, 1999]. Cette méthode de fusion a l'avantage d'être très performante mais elle nécessite un ensemble d'apprentissage conséquent. Son application à l'authentification biométrique est possible, pour peu que la base de données servant à construire le système soit de taille suffisante (plus de 50 utilisateurs).

1.2.4.6. Bilan sur la fusion

L'utilisation de la fusion est indispensable à notre avis dans tous les problèmes d'authentification biométriques. Les études d'autres chercheurs ont montré, à chaque fois qu'elle était correctement utilisée, des gains de performances importants. Un unique classificateur peut provoquer des erreurs très importantes s'il est utilisé seul, ses erreurs pouvant être composées facilement par d'autres classificateurs utilisés conjointement. L'utilisation de la fusion implique une attention particulière pour le choix des classificateurs et des caractéristiques biométriques. Il est crucial d'utiliser des classificateurs et des caractéristiques complémentaires si l'on souhaite que la fusion améliore les résultats. Le risque est de choisir deux classificateurs semblables (gaussienne et fenêtre de Parzen utilisant les mêmes données par exemple), provoquant alors des performances plus faibles que celle obtenue par le meilleur classificateur. Nous conseillons par ailleurs l'utilisation de l'opérateur *Somme* dans la majorité des cas. Lorsqu'on dispose d'une très grande base d'apprentissage l'utilisation d'une fusion par classificateurs devient également possible et peut donner de meilleurs résultats.

1.3. Bilan

Dans ce chapitre, nous avons présenté les différentes étapes d'une authentification biométrique ainsi que les problèmes de classification qui en découlent. Le problème de classification à une classe auquel nous sommes confrontés n'est pas encore résolu, aujourd'hui, de façon satisfaisante. Des auteurs ont proposé de nombreuses pistes pour s'affranchir des contraintes associées à ce type de problème pour les divers niveaux du processus mais bon nombre de problèmes restent encore mal résolus :

1. La sélection de caractéristiques a notamment fait l'objet de peu de recherches. Il est possible d'utiliser diverses méthodes lorsque les individus de la population partagent tous les mêmes caractéristiques. Il n'existe pas, aujourd'hui, de solution fiable pour sélectionner les caractéristiques quand les individus sont décrits par des caractéristiques différentes. Seules des heuristiques, dont la fiabilité reste à démontrer, sont alors utilisables.

2. Le filtrage des données biométriques bruitées est un problème encore largement étudié et pour lequel des solutions semblent satisfaisantes, dès que l'ensemble d'apprentissage est de taille suffisante. Dans ce cas, des algorithmes de classification non supervisée permettent d'éliminer assez facilement les observations problématiques. Des améliorations doivent encore être produites dans le cas des petits ensembles d'apprentissages

3. Les classificateurs disponibles aujourd'hui pour résoudre le problème de classification à une classe permettent d'approcher les performances des classificateurs à deux classes lorsque les conditions sont favorables. Pour obtenir des performances élevées, les classificateurs à une classe évolués (SVM, SVDD, 1-LVQ) nécessitent un ensemble d'apprentissage de grande taille (plus de 50 observations) et non pollué par des données bruitées. Quand l'ensemble d'apprentissage est de petite taille, les seuls classificateurs utilisables actuellement semblent être les plus rudimentaires (mesures statistiques ou mesures de similarité).

4. La phase de décision, notamment le choix du seuil de sécurité, prend une importance encore plus cruciale dans le cas du problème à une classe. La détermination d'un seuil sans disposer d'information sur la position des imposteurs par rapport aux utilisateurs authentiques est problématique ceci est d'autant plus vrai

pour la biométrie comportementale lorsque la stabilité des caractéristiques biométriques varie énormément suivant les utilisateurs.

5. Une fusion nous semble aujourd'hui indispensable, quel que soit le problème biométrique traité, le gain de performance constaté est important. Mais elle nécessite une réflexion lors du choix des classificateurs et du mode de fusion.

Le chapitre suivant reprend ces différentes problématiques et décrit comment nous proposons de résoudre chacun de ces verrous dans le cadre de la mise en place de systèmes d'authentification biométrique. Une originalité supplémentaire de notre travail est d'essayer de rester suffisamment générique pour que nos propositions ne soient pas spécifiques à un domaine d'application mais soient utilisables qu'elles que soient les données biométriques choisies.

Chapitre 2. Authentification biométrique : nos propositions

2.1. Introduction

Au cours de la mise en place d'un système d'authentification biométrique, une erreur est trop souvent commise : choisir et optimiser séparément chacune de ses composantes sans rechercher une cohérence d'ensemble. Cette cohérence est ordonnée par les impératifs que doit respecter le système, que ce soit au niveau de la sécurité, de la convivialité pour les utilisateurs ou de toute autre nature. Parfois, l'absence de réflexion globale peut aller jusqu'à causer l'oubli d'une des contraintes du système. Cette faille peut causer une dégradation catastrophique des performances du système et compromettre sa sécurité.

Au cours de notre travail, nous nous sommes intéressés principalement aux problématiques liées à la reconnaissance des formes, c'est-à-dire aux mécanismes permettant de décider si l'individu qui fournit la donnée biométrique peut être accepté ou non par le système. Les contraintes de sécurité liées, par exemple, à des problèmes de cryptage ou de stockage des données biométriques ne sont abordées que si elles ont une influence sur le processus permettant de prendre la décision.

Les contraintes pesant sur un système biométrique sont nombreuses, celles qui nous intéressent plus particulièrement imposent que le système :

- ait une « bonne » fiabilité : cette contrainte semble la plus évidente et la plus importante mais encore faut-il décider à partir de quel niveau de sécurité, la performance est satisfaisante. En fait, cela dépend de l'application et de l'objectif définis dans le cahier des charges.
- ne soit pas trop contraignant pour les utilisateurs, notamment lors des phases d'enregistrement et de reconnaissance. Là encore, les contraintes acceptables par un utilisateur dépendent de la population visée et de l'intérêt que les utilisateurs portent envers le système.
- fonctionne sur le « long terme » : Certaines caractéristiques biométriques évoluent au cours du temps. Dans ce cas, le système doit être capable de s'adapter à cette évolution.
- fonctionne correctement pour un maximum d'utilisateurs (idéalement tous). Le système doit pouvoir prendre en compte la diversité des comportements et des caractéristiques des utilisateurs.

Chacune de ces quatre contraintes a des conséquences sur la mise en place d'un système biométrique. Souvent les deux dernières sont négligées car elles n'apparaissent pas durant la phase de test, soit par manque d'utilisateurs, soit par manque de temps.

Notre contribution vise à proposer une architecture répondant à l'ensemble des contraintes précédemment citées. Pour cela, nous nous sommes basés sur une étude de cas particuliers puisque notre partenaire industriel désirait mettre en place un système d'authentification pour la dynamique de frappe. Néanmoins, nos propositions peuvent être utilisées dans de très nombreux problèmes traitant de la biométrie comportementale. Les principaux choix que nous avons faits peuvent être reconduits dans la plupart des problèmes de classification à une classe afin d'améliorer les performances de ce type de systèmes.

Dans ce chapitre, chaque étape entrant en jeu dans un processus d'authentification biométrique est revisitée afin de proposer des recommandations ou des améliorations aux méthodes utilisées. L'architecture que nous proposons est au préalable, décrite dans sa globalité alors que cette section s'achève par une discutions de nos méthodes et propositions pour personnaliser un système biométrique en fonction du comportement des utilisateurs.

2.2. Propositions pour l'architecture du système

L'architecture du système que nous proposons, représente un guide pour l'implémentation d'un système biométrique. Les étapes incorporées, nous semblent indispensables pour tous les problèmes d'authentification biométrique utilisant l'aspect comportemental. Par contre, dans le cadre d'utilisation de données biométriques physiques, certaines d'entre elles peuvent être optionnelles du fait de leur plus ou moins grande stabilité. Nous préciserons donc, au cas par cas, l'application de nos propositions sur les différents types de données biométriques.

2.2.1. Mise en place d'une base de référence

Les principales difficultés rencontrées dans le cadre de l'authentification biométrique comportementale proviennent de la variabilité des comportements des utilisateurs. Nous proposons de traiter cette variabilité par une plus grande adaptation du système à chaque utilisateur. Ce choix a fortement influencé la composition de notre architecture.

Malheureusement, comme bien souvent en biométrie, nous ne disposons pas dans le profil des individus de suffisamment d'information pour nous permettre de tirer des enseignements directement utilisables pour adapter le système. En effet, comme habituellement dans les problèmes de classification à une classe, la seule source d'information est l'utilisateur lui-même qui, au travers de la phase enregistrement, fournit aux systèmes des échantillons de sa donnée biométrique. Notre première proposition consiste donc à contourner ce problème en utilisant une deuxième source d'information que nous appelons la base de référence.

Ce que nous appelons base de référence est une base de données contenant des données d'utilisateurs acquises dans le cadre de la conception du système. Ces données sont obtenues avant la mise en service effective du système et contiennent les données brutes provenant d'utilisateurs représentatifs de la population visée ayant acceptés d'alimenter cette base de données. Cette base comporte, pour chaque utilisateur, trois types d'informations provenant directement des capteurs :

- des données qui proviennent de la phase d'enregistrement (profils),

- des données qui permettent de simuler un fonctionnement normal du système (reconnaissance),
- des données issues d'attaques simulées d'imposteurs.

Toutes les données sont labellisées ; il est donc possible de savoir pour chacune d'elles de quel utilisateur elles proviennent et si ce sont des données issues de la phase d'enregistrement (données d'apprentissage), des données acquises ensuite (reconnaissance) ou bien des attaques d'imposteurs. Les données de cette base, qui sont utilisées pour la conception du système ne peuvent donc pas être utilisées par la suite pour déterminer les performances du système final.

L'objectif de la base de référence est de faciliter les différents choix qui devront être faits pour chacune des étapes qui vont suivre. Elle permettra surtout la réalisation d'une personnalisation du système pour chaque utilisateur. Elle intervient donc à de multiples étapes de la conception et du fonctionnement du système.

Cette base de référence doit être assez fournie pour pouvoir représenter le plus fidèlement possible le panel des types d'utilisateurs auquel le système pourra être confronté lors de son utilisation normale. Le nombre d'utilisateurs présents dans celle-ci doit certes être important mais ceux-ci doivent également être assez diversifiés. Par exemple, si on travaille sur la reconnaissance vocale, la base de référence doit contenir des utilisateurs des deux sexes et ayant des timbres de voix différents et représentatifs. Si la base de référence est mal conçue, le système risque de connaître des dysfonctionnements avec des utilisateurs trop différents des individus de référence.

2.2.2. Recommandations pour la phase s'enregistrement

La création du profil à lieu pour chaque utilisateur à partir des données obtenues pendant la phase d'enregistrement. Nous présentons sur la Figure 13 l'architecture que nous préconisons pour cette phrase.

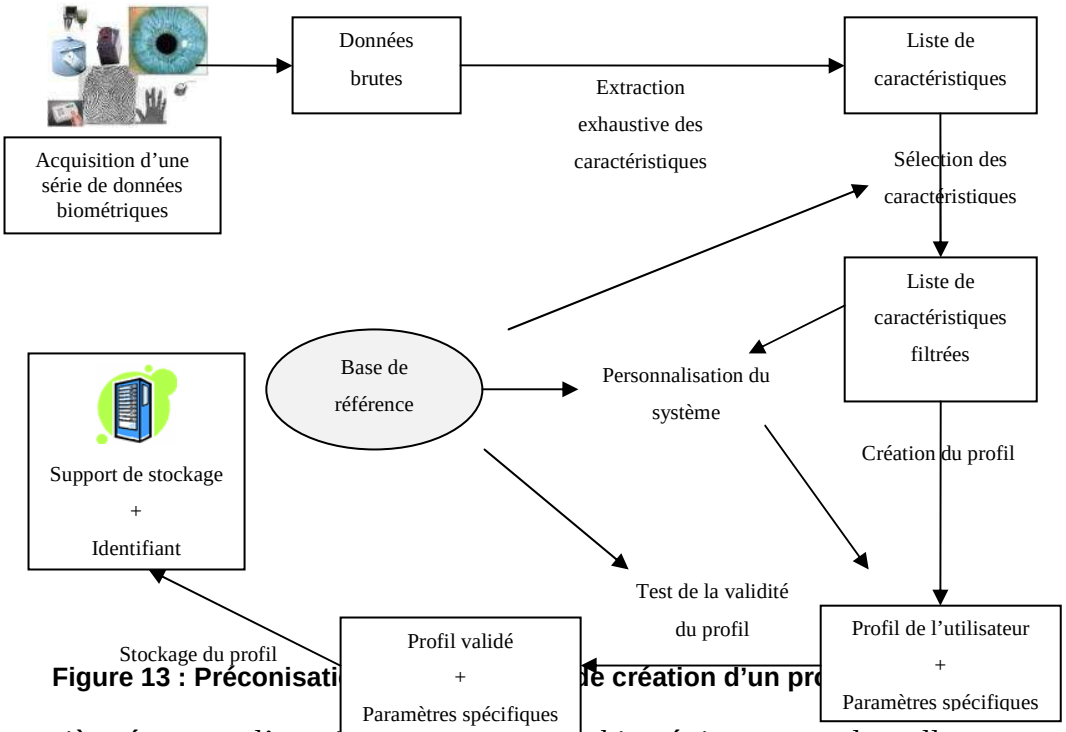


Figure 13 : Préconisati… de création d'un pr…

La première étape est l'acquisition des données biométriques pour laquelle nous avons déjà présenté les problématiques dans le premier chapitre.

De plus, cette phase, ainsi que la phase suivant, l'extraction des caractéristiques à partir des données brutes issues des capteurs, dépendent du choix des données biométriques utilisées. Nous ne traitons pas ces étapes car elles dépendent trop de la caractéristique biométrique utilisée.

Par contre, nous préconisons d'insérer systématiquement dans l'architecture une phase de sélection des caractéristiques afin d'optimiser le contenu du profil.

Déjà pour cette phase, l'intervention de la base de référence nous paraît opportune. Son utilisation procure des informations supplémentaires très importantes pour mieux sélectionner les caractéristiques, notamment les variances intra et inter classes des caractéristiques.

C’est également durant la création que certains paramètres utilisés par les classificateurs doivent être calculées.

Comme le montre la Figure 13, nous proposons la mise en place d’une phase de personnalisation du système. Cette personnalisation du système, également réalisée grâce à la base de référence, consiste en l’adaptation des paramètres des classificateurs à chaque utilisateur.

Nous proposons également d’ajouter à cette phase, une étape de test de la validité du profil obtenu : au cours de cette phase, la consistance du profil est évaluée afin de décider si celui-ci peut être utilisé sans risques dans la suite. En cas d’inconsistance du profil, un traitement spécifique doit être effectué : Il est possible, soit de redemander à l’utilisateur de rentrer les données biométriques, soit de modifier les contraintes d’admission des profils.

La dernière phase de la création du profil est le stockage du profil sur un support adéquat et la remise à l’utilisateur de son identifiant. Nous avons déjà abordé dans le Chapitre 1, les difficultés liées aux choix du lieu de stockage du profil et de sa forme. Nous ne présentons ici que les modifications liées à l’architecture globale du système, les détails de mise en œuvre sont fournis dans la suite.

2.2.3. Recommandations pour la phase de reconnaissance

L’architecture que nous proposons pour la phase de reconnaissance est résumée sur la Figure 14. En plus de prendre en compte la personnalisation du système, elle se caractérise par la mise en place d’un système de mise à jour permanent du profil. Cette première préconisation pour la phase de reconnaissance est souvent passée sous silence, voire inexistante dans les systèmes actuels, du fait qu’elle est surtout utile pour les systèmes comportementaux (encore peu utilisés actuellement) et beaucoup moins lors de l’utilisation de biométries physiques.

Les premières étapes de la reconnaissance, c'est-à-dire l’acquisition des données et la construction des vecteurs de caractéristiques, restent semblables à celles d’un système biométrique classique et à ce qui est fait durant la phase d’enregistrement.

Par contre, nous préconisons l'adjonction systématique d'une phase de fusion au module de reconnaissance. L'objectif est double : d'une part, regrouper les résultats de plusieurs classificateurs permet d'accroître les performances du système (voir chapitre 1), et d'autre part, la fusion constitue un levier afin d'effectuer une personnalisation du système à chaque utilisateur. La Figure 14 décrit les flux d'informations produits et utilisés par les différents classificateurs et autres modules pour produire une décision ou un score final. Dans cette étape, des paramètres adaptés à chaque utilisateur obtenus grâce au mécanisme de personnalisation sont fournis à chaque classificateur.

Ainsi, l'adaptation du système à chaque utilisateur implique le choix de paramètres personnalisés :

- lors de la création des scores de classifications : les paramètres sont alors utilisés par les classificateurs,
- lors de l'étape de fusion lorsque chaque méthode/ classificateur se voit affecter une pondération
- enfin, lors de la décision finale, c'est-à-dire par la définition de seuil de sécurité personnalisé.

La prise de décision est la phase finale de la phase de reconnaissance. Cette décision peut être l'acceptation de l'utilisateur ou son rejet et donc l'impossibilité pour lui d'accéder à la zone ou aux données protégées. Une attention particulière doit être apportée à cette phase, notamment au niveau du seuil de sécurité que nous proposons également de fixer localement pour chaque utilisateur.

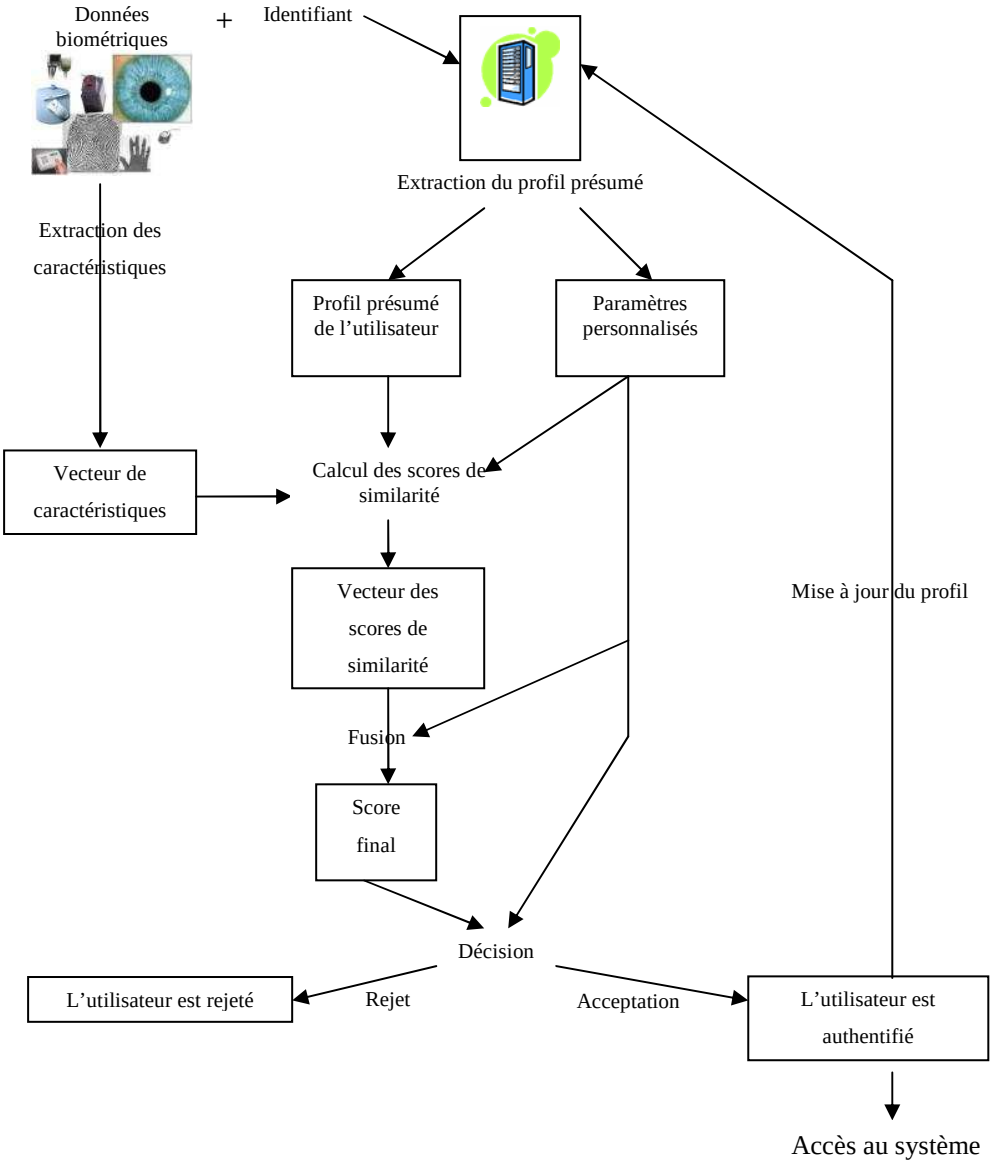


Figure 14 : Préconisation pour la phase de reconnaissance

Dans le cas où l'utilisateur est accepté, nous pensons indispensable l'implémentation d'une phase de mise à jour du profil surtout lorsque les données biométriques étudiées varient de manière non négligeable au cours du temps. Cette mise à jour consiste à prendre en compte, dans le profil de l'utilisateur, les données biométriques qui ont été validées au cours des phases successives de reconnaissance. Les profils sont ainsi régulièrement redéfinis afin de toujours correspondre au mieux aux comportements individuels des utilisateurs. Comme nous le verrons plus tard, ce choix n'est pas sans conséquences sur les classificateurs utilisables lors de la reconnaissance.

2.2.4. Préconisations pour l'évaluation de systèmes biométriques

L'évaluation d'un système biométrique se déroule généralement en deux temps : Dans un premier temps, les différents critères à utiliser pour évaluer le système sont déterminés, et dans un deuxième temps, les valeurs de ces critères dans une situation la plus proche possible de la situation réelle sont estimées.

2.2.4.1. Choix des critères pour l'évaluation du système

Le choix des critères d'évaluation du système doit se faire avec l'idée qu'ils permettront de vérifier quels sont les points forts et les points faibles du système biométrique qui est mis en place. C'est-à-dire vérifier si le système répond bien aux besoins initiaux. Ils doivent finalement permettre aux décideurs de faire le choix final à bon escient. Ces critères sont de deux types, quantitatif et qualitatif :

1. Les critères quantitatifs regroupent les critères de performances objectifs du système. Les critères les plus parlants sont les mesures de performances. Ces critères regroupent le TFA, le TFR, et le TEE (section 1.1.5). Pour mesurer fidèlement, les performances du système, nous conseillons de présenter non pas une valeur pour chacun des taux mais l'évolution du TFA et du TFR, en fonction du seuil de sécurité. Les valeurs des taux pour trois valeurs du seuil de sécurité doivent notamment être mises en lumière : les taux d'erreur pour un niveau de sécurité faible (TFR proche de zéro), pour un niveau de sécurité fort (TFA proche de zéro) et pour le niveau de sécurité au TEE. Les taux d'erreur pour les niveaux de sécurité fort et faible permettent de présenter le comportement du système aux configurations extrêmes. Le TEE permet de comparer le système d'authentification à d'autres systèmes biométriques existants. Nous estimons que d'autres critères numériques sont indispensables en complément des taux précédemment cités, ils concernent des indications sur la simplicité d'utilisation et la rapidité de fonctionnement du système. La valeur exacte de ces critères n'est pas forcément intéressante ; par contre leurs ordres de grandeurs donnent des informations essentielles sur la possibilité d'utiliser le système dans une situation réelle. En effet, l'utilisateur ne doit pas attendre un temps supérieur à quelques secondes, lors de la phase de reconnaissance avant d'avoir la réponse. Les temps intéressants que nous dégageons du système sont la

durée de reconnaissance, la durée de création du profil qui peut être plus longue que la durée de reconnaissance mais qui doit rester raisonnable. Le dernier temps est la durée de mise à jour du profil qui doit être, à notre avis de l'ordre de grandeur du temps de reconnaissance. Le dernier critère quantitatif, souvent négligé par les scientifiques, est le coût de l'implantation du système qui est une donnée essentielle pour les décideurs.

2. Nous considérons également essentiel d'ajouter des critères qualitatifs pour fournir des informations complémentaires bien que plus subjectives sur le système. Le but est de donner des informations non quantifiables sur le fonctionnement du système. Une des informations essentielles, doit indiquer comment les utilisateurs perçoivent le système. Cette perception intègre le taux d'intrusion perçue par les utilisateurs et le respect de leur vie privée. Une autre information importante est l'effort qui leur est demandée pour s'enregistrer ou être reconnu par le système. La présentation de ces données doit donc être réalisée pour tous les systèmes et permettre une comparaison avec les autres systèmes biométriques. Pour cela il possible d'attribuer une note à ces critères qualitatif, par exemple, à l'aide de notes sur une échelle de 1 à 10.

Les dernières informations essentielles que nous indiquons et qui dépendent étroitement du système sont les remarques particulières concernant le fonctionnement. Cela peut être des contre-indications : par exemple les empreintes digitales peuvent difficilement être utilisées dans un milieu où sont manipulées des substances abrasives. Ainsi, les facteurs pouvant causer une impossibilité d'utilisation doivent lorsque c'est possible être précisés (blessure au doigt pouvant altérer la dynamique de frappe…).

2.2.4.2. Détermination des critères

Une fois les critères d'évaluation déterminés, leur calcul, avant même l'installation définitive du système peuvent être problématique. Au contraire, certains des critères d'évaluation sont déductibles directement de la structure du système. Ces critères sont par exemple le coût de mise en place. De même, les efforts à demander à un utilisateur lors des différentes phases peuvent être obtenus par un rapide sondage auprès du public visé. Les questionnaires aux utilisateurs sont des outils performants qui permettent d'avoir rapidement des bonnes indications sur tout ce qui a trait à la perception du système par les utilisateurs.

La détermination des critères numériques, notamment ceux liés aux performances, nécessite l'utilisation d'une base de test. Cette base de test doit contenir des observations d'utilisateurs acquises dans des conditions le plus proches possibles du fonctionnement réel du système. Idéalement, cette base doit aussi comporter des observations d'imposteurs qui essaient de se faire passer pour des utilisateurs du système. Rappelons que dans nos préconisations une première base est déjà utilisée par notre système doit déjà remplir ces conditions : la base de référence. Bien évidemment, comme elle intervient à de nombreux points de la conception du système, elle ne peut pas être utilisée pour tester le système sous peine d'obtenir des performances estimées bien meilleures que celles qui seront réellement observées.

Pour obtenir une bonne indication des performances, il faut donc diviser les données acquises en deux bases :

- une fraction servira pour la base de référence
- et l'autre pour la base de test

Ce découpage des données peut être réalisé de nombreuses façons (tirage aléatoire des utilisateurs, utilisateur pair et utilisateur impair…). Le problème de cette division est qu'il est souvent difficile d'avoir assez de données pour constituer les deux bases, De plus si la division est mal faite, cela peut engendrer l'apparition d'un certain nombre de biais. Ces biais sont dus à quelques rares utilisateurs à problèmes qui sont responsables d'une grande partie des erreurs. Suivant leurs présences dans l'une des bases, les résultats peuvent être complètement différents si le nombre d'utilisateurs dans les deux bases est faible.

Pour limiter ce biais lorsque l'on dispose de peu d'échantillons de données, nous proposons d'utiliser une méthode d'estimation des performances baptisée validation croisée [Martens et Dardenne, 1998]. Avec cette méthode, pour calculer les performances du système pour un utilisateur *i*, les données sont divisées en prenant uniquement l'utilisateur *i* dans la base de test, et tous les autres utilisateurs dans la base de référence. Ce processus est répété pour tous les utilisateurs afin de produire les taux recherchés. Cette méthode donne une bonne estimation des performances réelles. Cependant, même en utilisant cette méthode, il est toujours préférable de récolter le plus d'informations, du plus grand nombre d'utilisateurs, afin d'avoir une estimation des performances la plus précise possible.

Une fois l'architecture du système établie, il faut choisir les composants qui interviendront à tous les niveaux. Même si le choix des composants dépend de l'application souhaitée pour les systèmes d'authentification, nous proposons une série de guides afin d'aider les personnes ayant à effectuer tous ces choix.

2.3. Propositions concernant les différents composants

2.3.1. Niveau de sécurité et exigence d'un système biométrique

Nous pensons que la première chose à faire, une fois la décision d'utiliser un système biométrique prise et avant même de commencer à choisir les différents éléments qui le constitueront, est de bien définir les contraintes auxquelles devra faire face le système biométrique.

Le choix des données biométriques doit être réalisé en fonction de ces contraintes. En effet, chaque type de données biométriques possède des avantages et des inconvénients qui mises en regard des contraintes indiquent son adéquation avec la situation.

Pour nous, la plus importante des contraintes qui pèsent sur la conception d'un système d'authentification biométrique est le rapport entre le budget qui peut être investi pour la mise en place du système et le degré de sécurité qui est souhaité. Il va de soit qu'il est impossible d'avoir une sécurité de très haut niveau avec un budget extrêmement réduit.

Le budget disponible est la plupart du temps imposé par des contraintes extérieures, pour décider de la ou les données biométriques à utiliser, il faut donc se concentrer sur le niveau de sécurité qui est exigé. Ce dernier est d'ailleurs bien souvent difficile à déterminer. En effet, quelle que soit la méthode biométrique choisie, nous estimons qu'il est toujours possible de tromper le système, il suffit d'y mettre le prix et de fournir les efforts nécessaires.

Le niveau de sécurité est donc déterminé par le coût et les efforts nécessaires pour casser le système. Il faut que ces efforts soient très supérieurs au gain que peut en espérer un individu cassant le système.

Même si nous ne nous intéressons pas beaucoup dans ce manuscrit aux aspects de la conception d'un système biométrique autres que ceux liés à la reconnaissance des formes, notamment aux aspects cryptage ou sécurité réseau, nous

estimons inutile que la sécurité des méthodes biométriques soit extrêmement élevée si celle des autres composants du système ne le sont pas. Dans tous les cas, un individu bien informé choisira de s'attaquer directement au point faible du système. C'est donc le niveau de sécurité du maillon faible qui définit alors la sécurité globale du système. Les données biométriques, les capteurs permettant de les acquérir, les supports de stockage et les techniques de cryptage doivent donc être choisis de façon à avoir un degré de sécurité du même ordre.

2.3.1.1. Le choix de la ou des données biométriques à utiliser

Suivant les contraintes de fiabilité et de prix, le premier choix concerne la sélection d'une des deux grandes classes de méthodes biométriques : les méthodes comportementales peu chères mais moins fiables, et les méthodes basées sur les caractéristiques physiques bien plus chères mais également bien plus fiables.

Les méthodes physiques nécessitent l'équipement de tous les lieux protégés avec des capteurs spécifiques qui sont souvent coûteux et parfois volumineux. Les méthodes comportementales nécessitent également des capteurs pour enregistrer les données biométriques étudiées, mais ceux-ci sont souvent peu chers ou bien déjà présents dans l'équipement de base du système à protéger.

Les données biométriques physiques sont stables au cours du temps et ne nécessitent donc pas forcément de mise à jour du profil. Par contre nous pensons les mises à jour indispensables dans le cas des méthodes qui étudient le comportement.

Les problèmes de classification biométrique peuvent souvent être traités comme des problèmes multi-classes dans le cas des données biométriques physiques. Cette possibilité simplifie considérablement les traitements car tous les utilisateurs utilisent la même séquence d'authentification. Les méthodes comportementales nécessitent, par contre, la plupart du temps, la résolution d'un problème de classification à une classe avec les contraintes associées (que nous avons déjà vues), notamment dues à l'utilisation de séquences d'authentification différentes pour chaque utilisateur.

Aux vues des contraintes associées à chacun des deux types de méthodes biométriques, nous préconisons l'utilisation des caractéristiques physiques plutôt pour protéger l'accès à des espaces physiques (par exemple des bâtiments, des salles…). Dans ce cas, car il faut de toutes façons rajouter une infrastructure matériel

afin de pouvoir procéder à la sécurisation de la zone. Par contre, pour l'accès à des ressources informatiques, les caractéristiques comportementales ont nettement l'avantage car des capteurs utilisables pour acquérir les données sont souvent déjà présents ou faciles à rajouter sur les dispositifs.

Une fois le type de biométrie choisie, la sélection des données elles-mêmes dépend des contraintes locales du système : présence ou non par défaut du ou des capteurs utilisés, fréquence d'utilisation, contraintes de l'environnement d'acquisition…. Nous ne nous attardons donc pas sur ces choix.

L'usage de la multi-modalité biométrique est de plus en plus fréquent. De nombreux chercheurs [Ben-Yacoub, 1999; Kumar *et al.*, 2003; Snelick *et al.*, 2003] ont montré que l'utilisation de plusieurs méthodes biométriques permettent d'améliorer considérablement les performances d'un système. Cette multimodalité pose néanmoins de nombreuses questions :

- Le choix des associations de caractéristiques biométriques est parfois difficile effectuer. Il est notamment intéressant de réfléchir sur la pertinence de coupler des caractéristiques physiques et comportementales. Souvent, il y a un grand écart de performances entre les caractéristiques biométriques comportementales et physiques. L'ajout d'une donnée comportementale à une donnée physique améliore probablement les performances du système. Cependant les performances des données physiques sont bien souvent déjà largement suffisantes et ce couplage rajoutent des contraintes très fortes liées à l'utilisation des comportements (problème à une classe, évolution au cours du temps…) sans en gagner les avantages (faibles coûts, convivialité…). L'effort demandé aux utilisateurs reste il supportable ? Difficile, de donner une réponse tranchée à cette question, néanmoins. Il ne semble alors pas judicieux d'utiliser conjointement des données issues des biométries comportementales et physiques dans une grande majorité des cas.
- Le couplage de deux méthodes biométriques physiques, augmente les performances, mais augmente également le coût d'installation du système, et surtout l'effort devant être fourni par l'utilisateur durant les phases d'acquisition. Nous conseillons donc le couplage de deux ou plusieurs méthodes seulement quand la demande de sécurité est très forte ou bien quand il est possible d'acquérir simultanément les deux caractéristiques utilisées (par exemple reconnaissance du visage et de l'iris à partir d'une seule photo d'un utilisateur).

- L'association de deux signatures comportementales peut aussi permettre d'augmenter les performances, principal point faible de ce type de méthodes. En effet, l'association de deux méthodes permet de disposer de plus de caractéristiques et donc à travers un module de sélection/combinaison de disposer plus facilement de caractéristiques discriminantes. Par exemple si on couple la dynamique de frappe et la reconnaissance de signature manuscrite, on peut espérer pour un utilisateur qu'une des deux caractéristiques biométriques sera stable. Nous pensons que cette association peut être profitable lorsque les dispositifs d'acquisition des caractéristiques couplées sont déjà tous deux présents sur le système. Le seul point négatif que nous voyons à coupler deux modalités comportementales est de contraindre l'utilisateur à faire un effort supplémentaire qui peut être important pour s'authentifier. Le paragraphe suivant propose justement une discussion sur les efforts qui peuvent être demandés à l'utilisateur durant l'enregistrement et la reconnaissance.

2.3.1.2. Que peut-on demander à un utilisateur ?

Nous pensons que le niveau d'acceptation par les utilisateurs est trop souvent sous estimés lors de la conception de systèmes. Cette acceptation dépend en grande partie de l'effort qui est demandé aux utilisateurs lors de l'enregistrement puis lors de la reconnaissance. La volonté de diminuer la quantité d'effort demandée aux utilisateurs se heurte à la nécessité de posséder suffisamment de données pour permettre une bonne reconnaissance. Dans le cadre du problème de classification à une classe, le volume de données nécessaire pour bien estimer la distribution des données dépend des classificateurs choisis, mais est très souvent important pour les plus performants d'entre eux, mais il est possible de le diminuer en éliminant certain type de classificateur (cf. Chapitre 1).

L'intérêt de réduire l'effort demandé à l'utilisateur ne se limite pas à son bien être : L'amélioration de la qualité des données fournies est également essentielle pour avoir de bonnes performances. Ce point est particulièrement crucial dans le cas de la biométrie comportementale. En effet, l'utilisateur s'authentifie grâce à son comportement, ce dernier doit donc être « naturel » pour espérer avoir un profil utilisable. Il faut donc éviter de solliciter un utilisateur fatigué, énervé ou stressé par le trop grand effort que lui est demandé. L'état d'esprit de l'utilisateur change son comportement et donc les données acquises ne lui correspondent plus.

Notons cependant, qu'au cours de l'enregistrement, l'effort demandé à l'utilisateur peut être plus important que lors de la reconnaissance car il n'a lieu qu'une seule fois. Cet effort doit, dans le cadre de la biométrie comportementale, simplement ne pas dépasser le stade ou l'utilisateur commence à être fatigué ou énervé. Une étude sur ce point, pendant la conception du système, en observant le comportement des utilisateurs, parait donc importante pour construire ensuite une base de référence pertinente.

Pendant la phase fréquente de reconnaissance, nous conseillons de limiter au maximum l'effort demandé. La quantité de données doit cependant rester assez importante pour permettre une bonne reconnaissance. Cette quantité doit selon nous être déterminée en fonction de la fréquence d'utilisation du système prévue, en fonction des performances désirées et en fonction de l'observation des utilisateurs lors des premiers tests. Dans le cadre de la biométrie comportementale par exemple, nous avons observé que des utilisateurs à qui on demandait de jouer à un jeu simple avec une souris ou d'entrer des séquences aux claviers, étaient totalement déconcentrés après moins de 5 minutes d'effort. Cinq minutes correspondent, à notre avis, à la durée maximale d'effort soutenu que l'on peut demander à un utilisateur. De même lors de l'authentification à l'aide de la dynamique de frappe, les tests ont montrés que dès qu'il fallait entrer plus de 3 fois une séquence pour la reconnaissance les utilisateurs désactivaient le système s'ils en avaient la possibilité. Ce qui donne une bonne indication sur la limite maximum de l'effort que l'on peut imposer aux usagers.

2.3.2. Construction du vecteur de caractéristiques

Nous regroupons sous cette dénomination, l'acquisition des données biométriques et les premiers traitements qu'elles subissent afin de produire les vecteurs de caractéristiques qui seront ensuite présentés aux classificateurs pour obtenir un score de classification.

Si les vecteurs de caractéristiques sont de mauvaise qualité, il n'est pas possible, même avec le meilleur classificateur, d'obtenir des taux d'erreur permettant un bon fonctionnement du système.

Les étapes que nous retenons dans cette phase sont détaillées sur la Figure 15. Dans la suite nous revenons sur chacun de ces modules. Au cours de la première phase, les caractéristiques biométriques sont calculées à partir des données bruitées fournies par les capteurs. Ensuite, une phase de sélection des caractéristiques les plus pertinentes permet de réduire les dimensions de l'espace des caractéristiques. Les caractéristiques sélectionnées sont normalisées puis regroupées en un ou plusieurs vecteurs de caractéristiques pour créer le profil ou demander l'authentification.

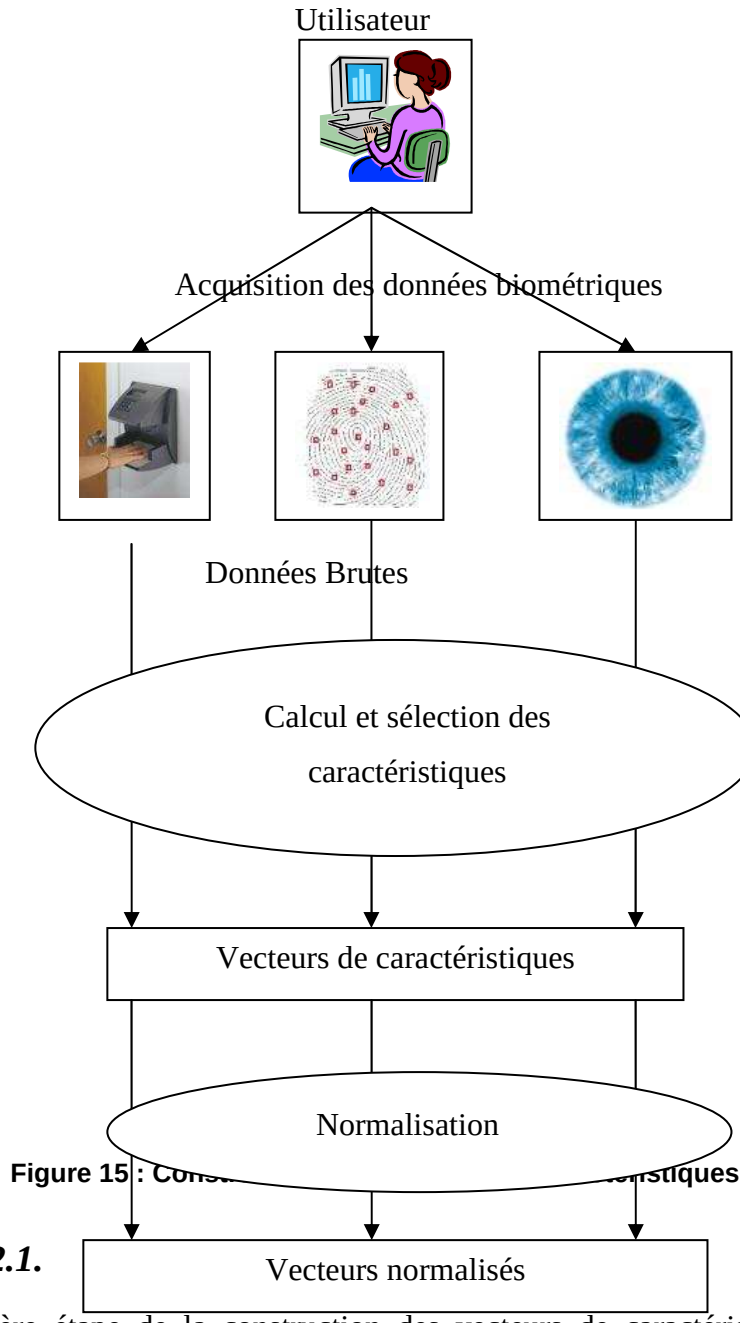


Figure 15 : Cons… …stiques

2.3.2.1.

La première étape de la construction des vecteurs de caractéristiques est l'acquisition des données biométriques. Cette étape met directement en contact

l'utilisateur et le système biométrique par l'intermédiaire des capteurs. Les problèmes à résoudre dans cette phase concernent bien sûr le choix des capteurs mais aussi de l'environnement d'acquisition.

Les caractéristiques des capteurs sont plus importantes qu'il n'y parait. L'utilisation de capteurs de qualités différentes peut perturber la reconnaissance. Nous préconisons donc l'uniformisation des capteurs, l'erreur à ne pas commettre est de choisir un capteur beaucoup plus performant pour la phase d'enregistrement en pensant ainsi améliorer la qualité du profil. Il y a dans ce cas, un risque d'introduire un biais qui rendra difficile les reconnaissances ultérieures.

Parfois, hélas, utiliser un même capteur pour l'enregistrement et pour tous les points d'accès n'est pas possible. C'est le cas par exemple, lorsque les capteurs ne servent pas uniquement aux acquisitions biométriques mais jouent également un autre rôle. Dans le cas de la dynamique de frappe, le clavier sert principalement au travail de l'utilisateur et est choisi pour le confort de l'utilisateur plutôt que pour les performances du système d'authentification. Lorsque différents capteurs sont utilisés, il faut alors faire en sorte que le système n'authentifie pas le capteur utilisé pour acquérir les données mais bien l'identité de l'utilisateur ! Au cours de la conception du système, nous préconisons donc de tester le plus grand nombre possible de capteurs possible en situation réelle afin de vérifier que le changement de capteurs ne détériore pas trop les performances.

L'autre grand facteur (perturbateur), devant être contrôlé attentivement, est l'environnement d'acquisition. Cet environnement peut avoir une grande influence sur les performances pour certaines méthodes biométriques. Par exemple, une mauvaise luminosité peut perturber la reconnaissance faciale ou de l'iris. Il est parfois difficile de contrôler l'environnement où seront pratiquées les acquisitions. Nous conseillons quand même de placer les capteurs dans un environnement stable se rapprochant le plus possible de celui rencontré lors de l'enregistrement. Si c'est impossible, nous préconisons de réaliser, lorsque cela est possible, des prétraitements sur les données afin de gommer une partie de l'influence de l'environnement. Par exemple, il peut être nécessaire de corriger l'éclairage sur une photographie.

Le dernier point auquel une grande importance doit être accordée durant le processus d'acquisition est l'utilisateur lui-même. Les facteurs que nous avons identifiés comme pouvant influencer le comportement de l'utilisateur sont nombreux. Ils peuvent être liés à son état d'esprit :

- comme nous l'avons déjà mentionné, s'il est énervé ou fatigué, son comportement peut être considérablement altéré.
- L'autre grande modification de comportement à laquelle nous avons attaché une attention particulière est l'habitude que l'utilisateur acquiert du système au fur et à fur mesure qu'il l'utilise. Cette habitude, une fois acquise, peut modifier progressivement les caractéristiques biométriques acquises. Même dans le cas de la biométrie physique, la qualité d'acquisition des données augmente au fur et à mesure du temps. Ceci nous entraîne à proposer, pour toutes les méthodes biométriques, un processus de mise à jour du profil, processus sur lequel nous revenons dans la suite.

Pour gommer tous ces effets, nous préconisons d'essayer de guider, le plus possible, l'utilisateur lors de l'utilisation du système. Pour cela, des guides physiques ou des notices explicatives détaillées constituent, par exemple d'excellents outils.

2.3.2.2. L'extraction et la sélection des caractéristiques

Souvent à partir des données fournies par le capteur, il est possible de produire un très grand nombre de caractéristiques différentes avec pour seule limite l'imagination des concepteurs du système. Le vrai problème, à notre avis, est l'évaluation de ces caractéristiques. Nous pensons qu'extraire un grand nombre possible de caractéristiques de différents types (numériques, discrètes, booléennes…), n'est pas la recette miracle pour obtenir de bonnes performances. Il est même possible, en ajoutant toujours plus de caractéristiques, d'obtenir des taux de performances catastrophiques. En effet, l'utilisation de « mauvaises » caractéristiques, c'est-à-dire des caractéristiques ne permettant pas de bien séparer deux utilisateurs, aura pour conséquence de faire décroître les performances.

La construction d'un bon vecteur de caractéristiques passe donc par la détermination des caractéristiques très discriminantes, c'est-à-dire permettant de bien différencier deux utilisateurs tout en restant stable pour un utilisateur donné. La détermination des caractéristiques les plus performantes pour séparer les utilisateurs se fait différemment en fonction de ce que l'on cherche :

- recherche de caractéristiques communes à tous les utilisateurs
- recherche de caractéristiques pouvant être différentes pour chaque utilisateur

Lorsque les caractéristiques sont communes à tous les utilisateurs, nous préconisons d'utiliser une base de référence lors de la phase de conception du système. Cette base permet, tous d'abord, de choisir les bonnes caractéristiques, déterminées à l'aide d'une fonction de performance associée à l'un des algorithmes de sélection de caractéristiques présentés dans le chapitre 1.

Chaque individu est unique donc les caractéristiques biométriques doivent différer pour tous les utilisateurs. Pour certaines méthodes biométriques, il n'y a pas le choix, les caractéristiques sont uniques par obligation (mot de passe différent dans la dynamique frappe par exemple). Le problème de la sélection de caractéristiques par des méthodes classiques est alors très difficile.

Il peut alors être tentant d'utiliser des heuristiques pour sélectionner les données les plus discriminantes. Ces heuristiques sont variables suivant les caractéristiques biométriques utilisées, elles utilisent souvent des connaissances a priori : par exemple pour l'étude de la dynamique de frappe d'un texte littéraire, nous connaissons les couples de touches les plus fréquent pour chaque langue, il peut donc être intéressant d'éliminer des caractéristiques pour les couples de touches rares voire très rares pour lesquels un utilisateur n'aura pas développé de style de frappe caractéristique. Ces heuristiques quand elles peuvent être mises en place permettent de réaliser une bonne première sélection des caractéristiques.

Une autre solution pourrait être de regarder les variances des caractéristiques. Celles dont les variances dans le profil sont les plus faibles pourraient, par exemple, être choisies mais rien ne prouve que ce choix soit judicieux. En effet, les caractéristiques ayant la plus forte variance chez un utilisateur peuvent être aussi celles qui ont la plus grande variance dans la population. Il nous semble donc plus intéressant d'examiner le rapport entre la variance des caractéristiques d'un utilisateur et celle de la population totale. Pour cela, il est à nouveau nécessaire d'utiliser, comme nous le préconisons, une base de référence en plus du profil de l'utilisateur.

2.3.2.3. La normalisation

La normalisation des données, simple à réaliser dans le cas classique, devient réellement problématique dans le cas du problème à une classe. En effet, la normalisation nécessite d'estimer les valeurs limites des caractéristiques inconnues lorsque seules les données contenues dans le profil de l'utilisateur sont disponibles.

Pour résoudre ce problème, nous proposons plusieurs solutions :

- La première d'entre elles est d'utiliser les valeurs théoriques des maximums, minimums, moyennes et écarts types quand celles-ci sont disponibles. Ces valeurs théoriques sont déterminées à partir des techniques utilisées lors de l'extraction des caractéristiques. Par exemple, le temps durant laquelle une touche d'un clavier est enfoncée ne peut être négatif. De même, au cours d'une séquence de frappe, ces temps ne peuvent normalement pas excéder une seconde. Le problème lié à l'utilisation de ces données théoriques ou heuristiques est qu'elles sont la plupart du temps très différentes de celles réellement observées. Nous conseillons donc de les utiliser uniquement quand il n'est pas possible de faire autrement.
- La base de référence peut également être utilisée pour déterminer les valeurs nécessaires à la normalisation lorsque les mêmes caractéristiques sont utilisées pour tous les utilisateurs. C'est d'ailleurs selon nous la meilleure solution.

Si chaque utilisateur utilise des caractéristiques différentes, la seule solution que nous pouvons proposer est d'estimer les valeurs limites à partir des données issues de l'enregistrement et stockées dans le profil. Le risque est alors d'avoir trop peu de données pour avoir une estimation fiable des valeurs limites. Même si ce risque est important, nous pensons qu'il est indispensable de normaliser les données pour un bon fonctionnement de la majorité des classificateurs.

2.3.2.4. Comment construire les vecteurs de caractéristiques ?

Une fois les caractéristiques, sélectionnées et normalisées, se posent les questions de leurs regroupements en un ou plusieurs vecteurs de caractéristiques, ainsi que le choix des classificateurs.

Une séparation des caractéristiques selon leur type peut être réalisée : les valeurs numériques (mesures de distance de temps….) d'un côté et les mesures discrètes (rangs, indices de classes…) de l'autre. En effet, il est rare que des classificateurs permettent de mixer différents types de variables sans perdre la spécificité de certaines d'entre elles.

Ce choix de découpage peut aussi dépendre des classificateurs. Les regroupements peuvent tenir compte du fait que certains classificateurs fonctionnent mieux avec des caractéristiques vérifiant certaines propriétés (au niveau du type comme vu précédemment, ou, par exemple, au niveau de la distribution des valeurs). Certains classificateurs nécessitent, par exemple d'avoir une distribution gaussienne des données pour obtenir de bonnes performances.

Le dernier questionnement sur ce sujet concerne la séparation ou le regroupement des caractéristiques provenant de différentes sources biométriques. Il ne nous semble pas exister de règles définitives pour résoudre cette question.

Par contre, nous conseillons, quand c'est possible, d'effectuer la sélection de caractéristiques en tenant compte des particularités et des contraintes de chaque classificateur pour obtenir de meilleures performances. De plus, nous avons vu que les algorithmes de sélection ne peuvent pas toujours être appliqués. Dans ce cas, le seul conseil que nous pouvons donner est de ne pas fournir trop de caractéristiques à un classificateur, notamment à cause de la petite taille des ensembles d'apprentissage dans les systèmes biométriques.

2.3.3. Stockage du profil

La première question qui se pose pour le stockage du profil concerne le choix du support de sauvegarde. Ce choix n'influe pas uniquement sur la sécurité et l'ergonomie du système, il influence le processus de reconnaissance en imposant bien souvent des contraintes de taille stockage. Nous avons déjà évoqué, dans le chapitre 1, la double possibilité :

- Un stockage centralisé des profils de tous les utilisateurs dans une unique base de données.
- Le stockage individualisé par l'intermédiaire d'un support amovible : (clé USB ou carte à puce par exemple).

Hormis les contraintes de sécurité du système et de protection de la vie privée, un stockage centralisé semble plus intéressant. Les coûts sont bien moindres puisqu'il ne nécessite qu'un unique support de stockage alors que pour un stockage individualisé, il faut ajouter le coût de mise à disposition des supports amovibles et veiller à l'installation de lecteurs à chaque endroit où les utilisateurs doivent pouvoir accéder au système.

Dans le cas d'un stockage centralisé, les utilisateurs ne peuvent pas égarer, endommager voire se faire voler leur identité biométrique. Ces comportements alourdissent, en effet, considérablement la gestion du système : il faut, à chaque fois, racheter le support, refaire l'enregistrement de l'utilisateur et prévoir une solution de remplacement le temps de la remise en service.

Par contre, pour la protection de la vie privée, un stockage centralisé est problématique. Un administrateur du système peut disposer d'informations critiques sur les utilisateurs et par des recoupements avec d'autres bases de données réussir à obtenir des informations confidentielles. Cette crainte reprise par la CNIL en France, limite considérablement les cas où des bases de données centralisées peuvent être utilisées. Le risque d'un stockage centralisé est également important en cas d'attaques ou de pannes. Si le stockage est compromis, tout le système est menacé et doit alors être réinitialisé (si c'est possible). Ce point est d'ailleurs moins problématique lorsque des données biométriques comportementales ont été choisies. L'utilisation d'un stockage centralisé semble alors plus intéressante.

Deux contraintes supplémentaires doivent être prises en compte lors du choix du support de stockage, l'une étant liée à des problèmes de sécurité et la deuxième aux performances du système de reconnaissance :

Il parait nécessaire de limiter au maximum les informations stockées dans le profil dans le but de limiter les risques pour la vie privée en cas de vol des informations biométriques. Limiter au maximum la taille des données incluses dans le profil réduit les coûts mais demande une réflexion supplémentaire pour déterminer quelles informations sont indispensables.

Si l'on ne désire pas de pertes de performances, il nous semble dangereux de trop limiter la quantité d'informations à stocker. Nous pensons indispensable de disposer de beaucoup d'informations, notamment pour la mise à jour du profil. Afin de pouvoir recréer les classificateurs à chaque nouvelle utilisation du système il est nécessaire de mettre à jour et d'inclure les dernières données acquises. Par contre, il est préférable de stocker uniquement les caractéristiques extraites et non les données brutes dans le but de limiter la possibilité de récupérer, reconstruire et voler les données biométriques personnelles.

Si la caractéristique biométrique choisie est basée sur des caractéristiques physiques, l'évolution des données est très lente, la mise à jour n'est alors pas indispensable et un stockage de taille réduite est alors possible, d'autant que dans ce cas, nous préconisons l'utilisation de supports individuels.

2.3.4. Choix des classificateurs

Le choix des classificateurs dépend des contraintes imposées et des connaissances a priori disponibles. Le choix du classificateur varie selon, d'une part, les types de données utilisées (numérique ou autres), et d'autre part, la quantité d'informations présentes dans l'ensemble d'apprentissage (profil et base de référence).

Avant de choisir un classificateur, nous conseillons d'examiner sa capacité à traiter les types de caractéristiques extraites. Si les caractéristiques sont non numériques, le choix des classificateurs est fortement restreint. Dans ce cas, il est possible d'utiliser des mesures de similarité adaptée, ou bien des classificateurs reconnus comme adaptés au traitement des variables non numériques (ART-1 pour les variables binaires par exemple). On peut également transformer ces données en données numériques mais leurs spécificités sont alors bien souvent perdues.

Un grand nombre de classificateurs est disponible si les caractéristiques sont exclusivement numériques ou si les données numériques sont traitées à part. Le choix du classificateur peut alors être guidé par des informations sur les types de distributions que suivent les caractéristiques. Si elles sont de type gaussien, l'utilisation de mixture de gaussiennes peut donner de bons résultats. S'il est possible de déterminer un modèle de la génération des données, alors il est possible d'utiliser une modélisation plus complexe, comme par exemple l'utilisation des chaînes de Markov cachées qui peuvent alors être utilisées de façon très efficace.

Le choix final peut dépendre de la connaissance des points forts et faibles des différents classificateurs adaptés aux problèmes de classification à une classe (voir chapitre 1). Par exemple, aujourd'hui les classificateurs basés sur les SVM à une classe semblent offrir des performances très intéressantes. Ils sont hélas peu efficaces lorsque le nombre de caractéristiques est important et qu'il y a peu d'observations dans l'ensemble d'apprentissage.

Le dernier facteur qui nous semble essentiel dans le choix du classificateur concerne le nombre de paramètres qu'il faut régler (plus ou moins finement et empiriquement) pour obtenir de bonnes performances. Ces paramètres peuvent être par exemple dans le cas des réseaux de neurones le nombre de couches, le nombre de neurones par couche, ou le nombre de voisins examinés pour les k-ppv, … En authentification biométrique, le choix de ces paramètres est délicat. En effet, dans le cas général, aucune donnée d'imposteurs n'est disponible pour évaluer les performances en fonction des paramètres choisis. C'est pourquoi nous préconisons d'utiliser une base de référence afin de les fixer au moins de façon globale, dans un premier temps. Dans le cas des méthodes comportementales, les variations de ces paramètres peuvent avoir des conséquences très importantes suivant les utilisateurs ou les conditions d'acquisition. Cela peut aller jusqu'à l'impossibilité pour certains usagers d'utiliser le système. Malheureusement, fixer les paramètres des classificateurs de manière adaptée à chaque utilisateur est une tâche compliquée. Nous présentons, dans la suite, des méthodes facilitant la détermination des jeux de paramètres mais celles-ci deviennent moins fiables lorsque le nombre de paramètres augmente. Durant nos expérimentations, nous avons noté que certains classificateurs sont très sensibles au choix des paramètres (par exemple les mixtures de gaussiennes), et nécessitent un réglage très fin ; une petite variation des paramètres peut faire chuter complètement les performances du classificateur.

En résumé, pour choisir les classificateurs, nous utilisons d'abord les connaissances a priori sur le problème à résoudre et les types de caractéristiques à traiter. Si ces informations ne nous permettent pas de choisir parmi les classificateurs disponibles, la sélection doit se faire suivant le contenu et la taille de l'ensemble d'apprentissage. Dans le cas d'un ensemble d'apprentissage de grande taille avec des vecteurs de caractéristiques de dimension raisonnable, nous conseillons les SVDD [Tax et Duin, 1999] ou les SVM à une classe [Schölkopf *et al.*, 2001] qui sont aujourd'hui les classificateurs les plus performants. A l'opposé, si l'ensemble d'apprentissage est très limité, nous conseillons l'usage de classificateurs plus simples : mesures de similarité ou méthodes statistiques. De plus, lorsque c'est possible, nous préconisons d'utiliser plusieurs classificateurs de différents types. Les performances augmentent souvent significativement au travers d'une phase de fusion. Nous conseillons également de modifier les classificateurs afin qu'ils donnent un score plutôt qu'une décision. C'est en général assez simple, il suffit de supprimer la phase de seuillage, et de faire en sorte qu'ils génèrent une distance. Pour les classificateurs complexes, comme les SVM, il est possible d'obtenir un score en prenant en compte, par exemple, la distance à la marge. Pour les SVDD, la distance au centre de la sphère qui englobe les données peut être utilisée.

2.3.5. Fusion

Dans cette partie, nous détaillons la phase de fusion qui correspond à une étape importante de notre architecture. Quand plusieurs classificateurs sont disponibles, nous mettons en place une procédure de fusion afin de construire le score final. L'intérêt de la fusion de classificateurs a été présenté dans le chapitre 1. Nous préconisons d'utiliser une méthode de fusion de classificateurs par combinaison des scores et plus particulièrement l'opérateur *Somme*. Cet opérateur est généralement considéré comme l'un des plus performants et également l'un des plus simples à utiliser. Un autre avantage considérable de l'opérateur *Somme* est la possibilité de pouvoir facilement ajouter une pondération aux différents classificateurs. L'objectif de cette pondération est de prendre en compte la différence de performance des classificateurs individuellement pour chaque utilisateur. Nous les adaptons à chaque utilisateur car nous pensons qu'un classificateur peut être plus adapté à un type de comportements qu'à un autre. Les poids qui sont affectés à chacun d'entre eux doivent donc également varier.

Nous proposons donc l'adaptation de l'opérateur *Somme* aux problèmes de classification à une classe de la façon suivante :

Nous appelons $Score^i$, le score donné par le *ieme* classificateur parmi les N utilisés. Le problème à résoudre est un problème d'authentification et non d'identification les règles de fusion définit par Kittler [Kittler *et al.*, 1998] doivent donc être modifiées :

La première modification a été de considérer notre problème comme un problème à deux classes en posant l'équation (13).

$$P(acceptation) = 1 - P(rejet) \quad \textbf{(13)}$$

Comme nous ne pouvons avoir accès aux probabilités mais disposons de scores normalisés pour être compris dans l'intervalle [0,1], nous remplaçons les probabilités en utilisant l'équation (14).

$$P(acceptation|i) = Score^i \quad \textbf{(14)}$$

Nous n'avons pas accès à des les probabilités a priori de répartitions observations entre imposteurs et utilisateur authentiques du fait de la nature même du problème, nous allons directement calculer le score final à l'aide de l'équation (15).

$$ScoreFinal = \sum_{i=1}^{N} Score^i \quad \textbf{(15)}$$

La pondération est réalisée par l'intermédiaire d'un poids w_i associé au classificateur *i.* Le score final est alors calculé par l'équation (16) .

$$ScoreFinal = \sum_{i=1}^{N} w_i * Score^i \quad \textbf{(16)}$$

Les poids peuvent être fixés de façon globale suivant les performances observées sur la base de référence, mais nous proposons de les individualiser pour chaque utilisateur. Ils sont alors déterminés lors de la phase d'estimation des paramètres (voir les préconisations sur l'architecture de la phase d'enregistrement). Ils permettent ainsi de mieux personnaliser le système en indiquant quels sont les types de caractéristiques les plus performantes pour chaque utilisateur.

La fusion permet donc, d'une part, de prendre en compte plusieurs classificateurs et, d'autre part, d'adapter encore plus le système à chaque utilisateur et ce d'une façon la plus simple et efficace possible.

2.3.6. Préconisations pour la phase de décision

Une fois le score final obtenu, il reste encore une étape importante : prendre la décision finale. Cette décision résulte de la comparaison du score avec le seuil de décision.

Ce seuil peut être défini de façon globale à partir de la base de référence et fixée à l'identique pour tous les utilisateurs. Néanmoins, cette façon de fixer le seuil de sécurité n'est pas satisfaisante. En effet, il existe une très grande disparité entre les variabilités des profils des différents individus surtout dans la biométrie comportementale. Nous proposons de résoudre ce problème de disparité en fixant un seuil de sécurité par individu. Les détails sur les moyens utilisables pour effectuer le choix du seuil de sécurité sont présentés dans la partie « Personnalisation des paramètres ».

Même si nous pensons préférable de déterminer un seuil pour chaque individu, nous pensons qu'il est important de laisser la possibilité aux administrateurs du système de régler ce seuil afin de permettre une meilleure adaptation du système en cas de problème.

Dans l'idéal, un réglage manuel du seuil de décision doit permettre de passer d'un taux d'erreur correspondant à un TFA de 0% à un taux d'erreur correspondant à un TFR de 0%, afin que les utilisateurs du système puissent le régler pour passer d'une sécurité maximal à une tolérance maximal. Pour simplifier la tâche de l'administrateur du système nous proposons de définir une interface d'administration permettant de multiplier le seuil de décision issu de l'étape de personnalisation des paramètres par un cœfficient multiplicateur ε. Pour simplifier le réglage, il est possible de déterminer les valeurs de ce cœfficient pour différentes configurations de sécurité sur la base de référence. Ces configurations de sécurité incluent :

- le cœfficient pour le TEE
- le cœfficient pour une sécurité maximale (TFA=0%)
- le cœfficient pour une utilisation conviviale (TFR =0 %)…

Les administrateurs peuvent ensuite choisir le ε qui correspond le mieux au fonctionnement souhaité ou choisir des valeurs proches des valeurs préconfigurées en fonction de leur besoin.

2.4. Authentification biométrique individualisée

On l'oublie trop souvent, un système biométrique est conçu pour fonctionner avec des utilisateurs. Ceux-ci sont tous différents et ont des caractéristiques distinctes (celles qui nous permettent de les authentifier notamment…). Un système biométrique comportemental considérant tous les utilisateurs comme provenant « d'un même moule », fonctionne mal.

Ainsi, selon nous, lorsque la biométrie comportementale est utilisée le système doit obligatoirement s'adapter à chacun des utilisateurs. Les procédés d'adaptation sont les suivants :

- Individualisation des paramètres des classificateurs : les moteurs d'authentification utilisent un certain nombre de paramètres. Nous proposons de rendre propres à chaque utilisateur ces paramètres généralement fixés de manière identiques pour tous les individus,
- Evolutivité des paramètres : le système doit suivre l'évolution de l'utilisateur au cours du temps en mettant régulièrement à jour son profil.
- Evaluation, dès l'enregistrement, des données biométriques acquises pour détecter les profils problématiques ou incompatibles. Cette décision doit permettre de détecter les utilisateurs à problème à cause desquels des imposteurs pourront tromper le système. Cette décision ne se traduit pas forcément par le refus d'enregistrement d'un utilisateur, mais oblige celui-ci à refaire le processus d'acquisition ou à modifier ce qui le caractérise (changer de doigts par exemple pour la reconnaissance par empreintes digitales, changer de signature pour la reconnaissance de signature manuscrite, changer de mot de passe pour la dynamique de frappe…).

Toutes ces méthodes que nous proposons pour mettre en œuvre une personnalisation du système sont décrites dans la section suivante.

2.4.1. Personnalisation des paramètres du moteur d'authentification

En biométrie, et plus particulièrement en biométrie comportementale, les paramètres des classificateurs jouent un rôle important sur la performance du système. L'utilisation de paramètres globaux c'est-à-dire communs à tous les utilisateurs, pose de grands problèmes car les comportements des utilisateurs et leur évolution dans le temps est très importante. Nous proposons donc de définir un utilisateur non seulement par un ensemble de vecteurs de caractéristiques, mais également par un jeu de paramètres adaptés. Un des objectifs est d'éviter le cas d'utilisateurs ayant un profil tel que toutes les observations soient acceptées (quelles viennent de l'utilisateur ou pas), ou bien à l'inverse qu'elles soient toutes refusées (y compris celles provenant réellement de l'utilisateur).

Quelques tentatives d'individualisation ont déjà eu lieu sur des systèmes biométriques multimodaux. Par exemple, dans [Jain et Ross, 2002] et [Fierrez-Aguilar *et al.*, 2005], les seuils de décisions et les poids utilisés durant l'étape de fusion des différentes modalités biométriques sont fixés séparément pour chaque utilisateur. Le problème de ces tentatives vient de leurs méthodes de calcul qui nécessitent d'employer des données d'autres utilisateurs ou d'imposteurs normalement non disponibles dans les applications réelles. En effet, les auteurs fixent les divers paramètres de manière à minimiser une erreur de classification (Taux de mal classé ou TFA et TFR) qui ne peut normalement être obtenue que si on dispose, en plus de l'ensemble d'apprentissage de l'utilisateur, d'observations fournies par le même utilisateur et de données d'imposteurs. Dans les applications réelles du problème à une classe, nous n'avons pas la possibilité d'avoir accès à ces données, la détermination de ces paramètres locaux est alors bien plus compliquée.

Nous avons déjà indiqué qu'une base de référence pouvait être utilisée dans le but de déterminer les paramètres individuels d'un utilisateur (ne faisant pas partie de la base de référence). Avec cette solution, il n'est pas possible d'atteindre les performances des méthodes utilisant des données d'imposteurs « réels » car nous avons à notre disposition bien moins d'information. Nous pouvons cependant espérer faire mieux qu'avec des paramètres identiques pour tous les utilisateurs, notamment

pour ceux qui posent de graves problèmes de performances à cause de la dispersion des caractéristiques composant leur profil.

Avant de présenter les méthodes que nous utilisons pour l'estimation des paramètres individuels, il est nécessaire de déterminer quels sont les paramètres à fixer globalement et ceux devant être propres à chaque utilisateur. Deux indications peuvent être utilisées pour faire ce choix :

- L'influence du paramètre sur les performances : en cas de très faibles variations des performances, il n'est pas nécessaire, d'alourdir le système en mettant en place une méthode d'adaptation.
- L'intervalle de variation des valeurs pouvant être attribuées au paramètre (selon les utilisateurs). Grâce à la base de référence, il est possible de réaliser des tests afin d'essayer de déterminer les bonnes valeurs des paramètres pour chacun des individus de référence. Ces valeurs peuvent être aussi déterminées manuellement ou en utilisant des heuristiques. Si l'intervalle de variation des « bonnes valeurs » du paramètre est important, une adaptation locale s'impose.

Ainsi, les paramètres que nous décidons d'adapter localement sont ceux dont les valeurs efficaces varient suivant les utilisateurs et dont la modification joue un rôle important sur les performances du système.

Pour estimer les paramètres, nous avons conçu une architecture qui est présentée Figure 16. Le principe de cette architecture est de créer des estimateurs qui déterminent le jeu de paramètres individuels à partir des données obtenues lors de la phase d'enregistrement et à l'aide uniquement de la base de référence. La création de ces estimateurs se base sur des caractéristiques différentes de celles utilisées pour la reconnaissance. Nous regroupons l'ensemble des caractéristiques utilisées pour obtenir les estimateurs dans un vecteur que nous appelons le vecteur *Ref* et qui est propre à chaque profil. Le vecteur *Ref* est complémentaire des caractéristiques biométriques contenues dans le profil. Nous présentons dans la suite quelques pistes pour déterminer ce vecteur. Les paramètres recherchés sont également regroupés dans un vecteur que nous appellerons *Param*. La difficulté est de trouver un lien entre les vecteurs *Ref* et les vecteurs *Param* contenant les paramètres.

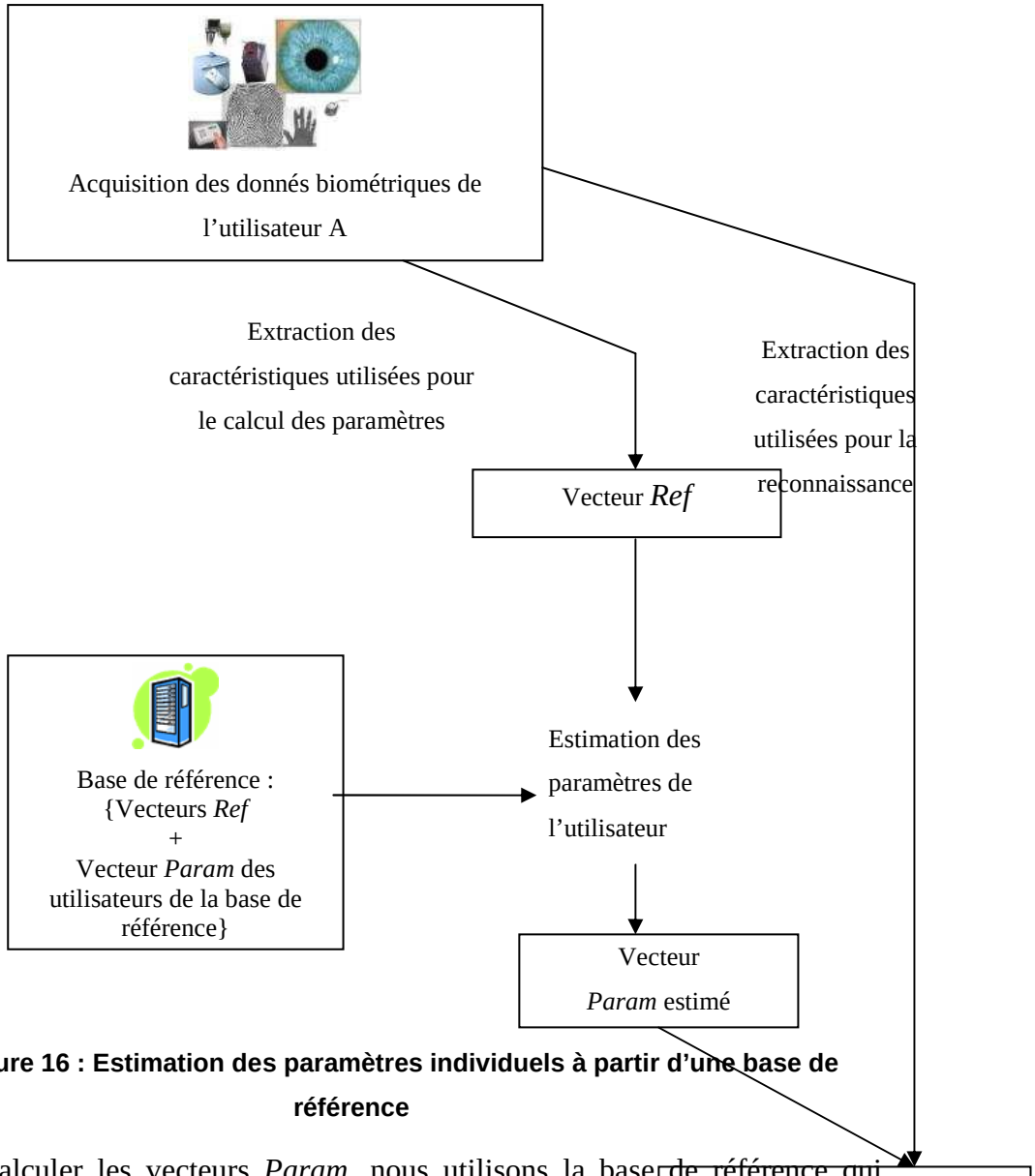


Figure 16 : Estimation des paramètres individuels à partir d'une base de référence

Pour calculer les vecteurs *Param*, nous utilisons la base de référence qui contient des données d'utilisateurs et des attaques d'imposteurs. Pour chaque utilisateur, nous déterminons le jeu de paramètres optimal en réalisant une exploration exhaustive de l'espace des paramètres. Les paramètres retenus sont ceux qui minimise la somme TFR plus TFA, les taux sont calculés à l'aide des données des imposteurs et des d teurs présents dans la base de référence. Nous avons choisi d'uti exhaustive dans l'espace des paramètres, cela peut prendre beauc ant les classificateurs choisis (de quelques minutes à plusieurs jours en fonction du mode d'exploration de l'espace) mais cette détermination n'a lieu qu'une seule fois lors de la conception du système.

Ensuite, solutionner le problème de l'estimation des paramètres, revient à résoudre deux problèmes spécifiques :

- Le premier problème est la sélection des caractéristiques qui permettront de créer les fonctions d'estimations. Ces caractéristiques sont regroupées dans les vecteurs *Ref*.
- Le second problème est la création des fonctions d'estimation qui détermineront les paramètres locaux à partir des vecteurs *Ref* extraits sur les profils.

Nos solutions à ces deux problèmes sont présentées dans la section suivante.

2.4.1.1. Construction du vecteur Ref

La première difficulté de l'estimation des paramètres, correspond à la détermination des caractéristiques servant à estimer les paramètres locaux. Ces caractéristiques sont obtenues à partir des informations disponibles pour un utilisateur une fois la phase d'enregistrement effectuée. Les caractéristiques qui peuvent être extraites sur un profil varient énormément suivant le problème biométrique traité mais aussi suivant les paramètres à estimer. Par exemple, pour déterminer un seuil de décision, les caractéristiques performantes sont celles qui donnent une indication sur la variabilité du profil. Les rapports moyenne/variance des différentes caractéristiques sont un exemple de caractéristiques utilisables.

Les scores donnés par les différents classificateurs utilisés lors de l'étape d'authentification sont, selon nous, des caractéristiques extrêmement intéressantes pour l'estimation des paramètres, est basées sur le calcul. En effet, leur moyenne et variance peuvent donner des informations sur les poids à affecter à chaque classificateur lors de la phase de fusion ainsi que sur le seuil de décision.

Pour calculer ces scores, nous ne disposons que de l'ensemble d'apprentissage et la base de référence, il faut donc utiliser une procédure particulière pour calculer ces scores, dérivé de la validation croisée. Les classificateurs utilisés pour calculer les scores sont créés avec des paramètres globaux déterminés grâce à la base de référence, et à l'aide de l'ensemble d'apprentissage de l'utilisateur auquel on retire l'observation dont on cherche à calculer le score.

Bien d'autres caractéristiques extraites des profils peuvent être exploitées mais toutes dépendent étroitement du problème. Citons pour exemple, dans le cas de la dynamique de frappe, le temps moyen pour frapper un mot de passe au clavier et la variance de ce même temps.

2.4.1.2. Procédures d'estimation des paramètres individuels

Les procédures d'estimation des paramètres propres à chaque utilisateur que nous proposons fonctionnent grâce à la mise en place d'une base de référence. Les performances de ces différentes méthodes seront présentées sur un problème spécifique dans la troisième partie de ce manuscrit.

Nous dégageons deux grandes façons de procéder afin d'estimer les paramètres locaux :

- L'estimation directe de la valeur des paramètres à partir des vecteurs *Ref* définis préalablement : L'objectif est alors de créer une fonction qui à partir d'un profil, donne directement les valeurs adaptées des paramètres.
- La division des utilisateurs en classes partageant le même comportement (clustering). Cette méthode exploite un algorithme de regroupement des profils à partir des vecteurs *Ref*. Les paramètres affectés à l'utilisateur correspondent à ceux associés à la classe à laquelle a été affecté son profil. Deux variantes peuvent être mises en place puisque les classes de comportement peuvent être construites à partir des vecteurs *Ref* ou bien à partir des jeux de paramètres optimaux calculés sur la base de référence.

2.4.1.2.1. *Estimation directe*

La méthode le plus simple qui puisse être utilisée pour affecter un jeu de paramètres à un profil, consiste à utiliser le jeu de paramètres optimaux d'un autre profil appartenant à la base de référence. L'intérêt de cette méthode est sa simplicité et sa performance qui peut être très bonne si la base de référence est représentative de la population.

Pour décider, quel jeu de paramètres associer à l'individu, il suffit de calculer les distances entre le vecteur *Ref* du profil de l'utilisateur, et tous les vecteurs *Ref* des profils de la base de référence. Le jeu de paramètres retenu sera celui du plus proche voisin dans la base de référence.

Le point faible de cette méthode est sa très forte sensibilité : le jeu de paramètres dépend uniquement d'un seul profil, il suffit que les paramètres de ce profil aient été mal déterminés, ou bien qu'il n'y ait pas dans la base de référence de profil suffisamment semblable, pour que les performances soient catastrophiquement dégradées.

Pour réduire cette sensibilité nous proposons de prendre en compte les trois voisins les plus proches au lieu d'en utiliser seulement un. Le jeu de paramètres affecté au profil correspond à la moyenne des jeux de paramètres des trois profils voisins. Le risque d'instabilité de la méthode diminue considérablement, tout en conservant une bonne adaptation à chaque profil comme le montre les expérimentations que nous avons menées et qui seront présentées dans le chapitre suivant.

D'autre estimateurs très performants et fonctionnant même dans le cas de problèmes non-linéaires peuvent être utilisés.

Nous avons testé un perceptron multicouches avec comme algorithme d'apprentissage : *Resilient Backpropagation* (RPROP)([Riedmiller et Braun, 1994]. Nous avons choisi ce classificateur car le perceptron multicouche obtient en général de bonnes performances sur les problèmes d'estimation.

Ce réseau de neurones dispose d'autant d'entrées qu'il y a de valeurs dans le vecteur *Ref*, et d'autant de neurones de sorties qu'il y a de paramètres à estimer.

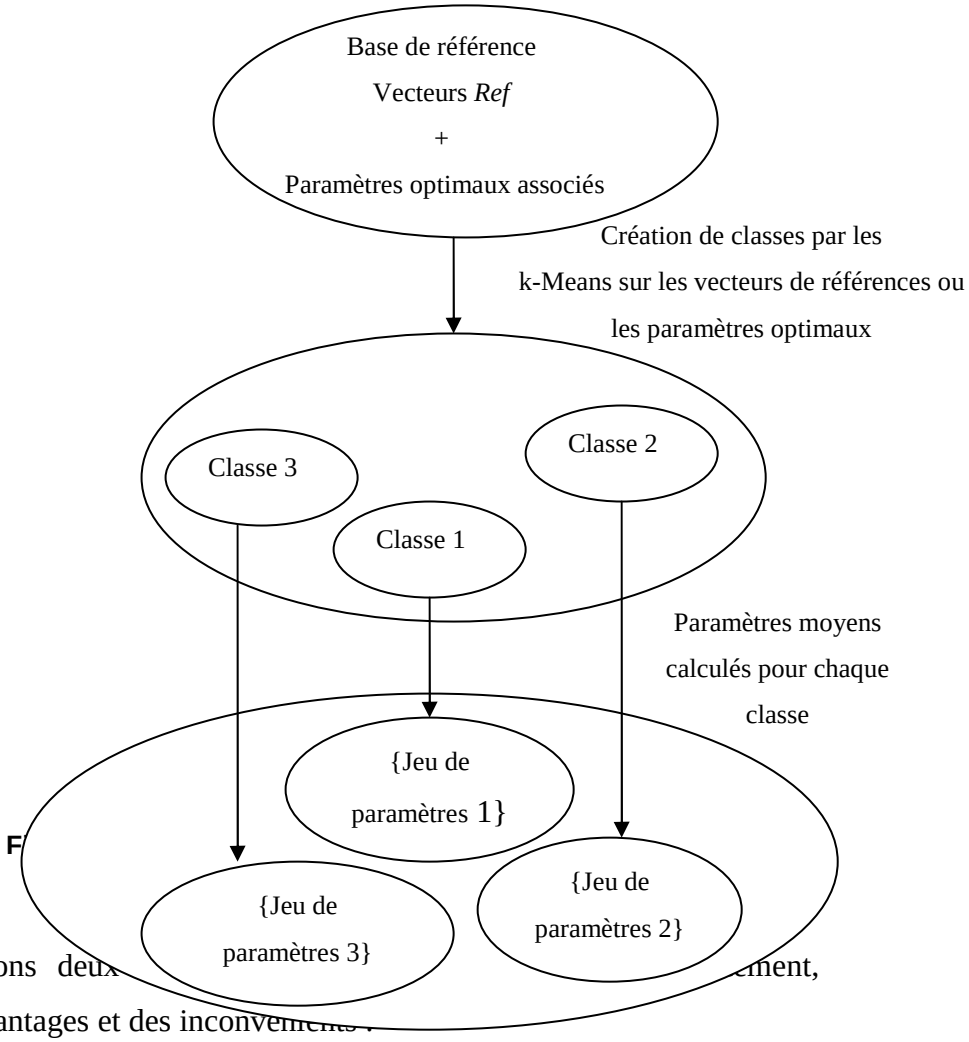
Au vu du problème à résoudre, le nombre de couches cachées, peut être fixé à 1 afin d'éviter un sur-apprentissage dû à la taille de la base de référence (souvent assez faible). Ce réseau de neurones est entraîné en fournissant en entrée les vecteurs *Ref* extraits des profils de la base de référence, et en comparant la sortie avec les paramètres optimaux associés à ces profils, les poids du réseau de neurones sont mis à jour à l'aide de l'algorithme RPROP.

Il nous semble que l'utilisation du réseau de neurones est préférable à l'utilisation d'une simple régression car ils sont réputés plus performants et permettent donc des estimations plus complexes.

2.4.1.2.2. *Création de classes de « comportement »*

Estimer directement les paramètres d'un profil est un problème difficile. Il n'est pas toujours possible de mettre au point un estimateur fiable fournissant directement les paramètres personnels d'un individu. Nous pensons que les

utilisateurs ayant des comportements proches doivent avoir des paramètres fonctionnels proches. Nous proposons donc de simplifier le problème en créant des classes « types » d'utilisateurs au sein de la base de référence. Ces classes regroupent les profils d'utilisateurs dont nous avons observé qu'ils avaient le même comportement lors de la phase d'enregistrement. Un jeu de paramètres adaptés est associé à chacune de ces classes de comportement (Figure 17).



F

Nous proposons deux ment,
chacune ayant des avantages et des inconvénients :

1. La première solution consiste à regrouper les profils suivant une mesure de similarité. Les classes sont déterminées suivant la proximité entre les vecteurs *Ref*. Nous utilisons l'algorithme de classification non supervisé k-means [McQueen, 1967] pour créer les classes sur l'ensemble des vecteurs *Ref* de la base de référence.

Une fois les classes créées, il faut déterminer le jeu de paramètres à associer à chacune d'entre elles. Pour cela, nous avons choisi de calculer la moyenne des jeux de paramètres des profils des individus affectés à la même classe.

Les k-means nécessite de régler le paramètre k (nombre de classes). Ce paramètre ne peut être identique pour tous les problèmes, le nombre de classes dépend notamment du nombre de paramètres à estimer et de la dispersion des points représentant les vecteurs *Ref* dans l'espace. Le choix du nombre de classes ne peut donc être réalisé qu'au cas par cas.

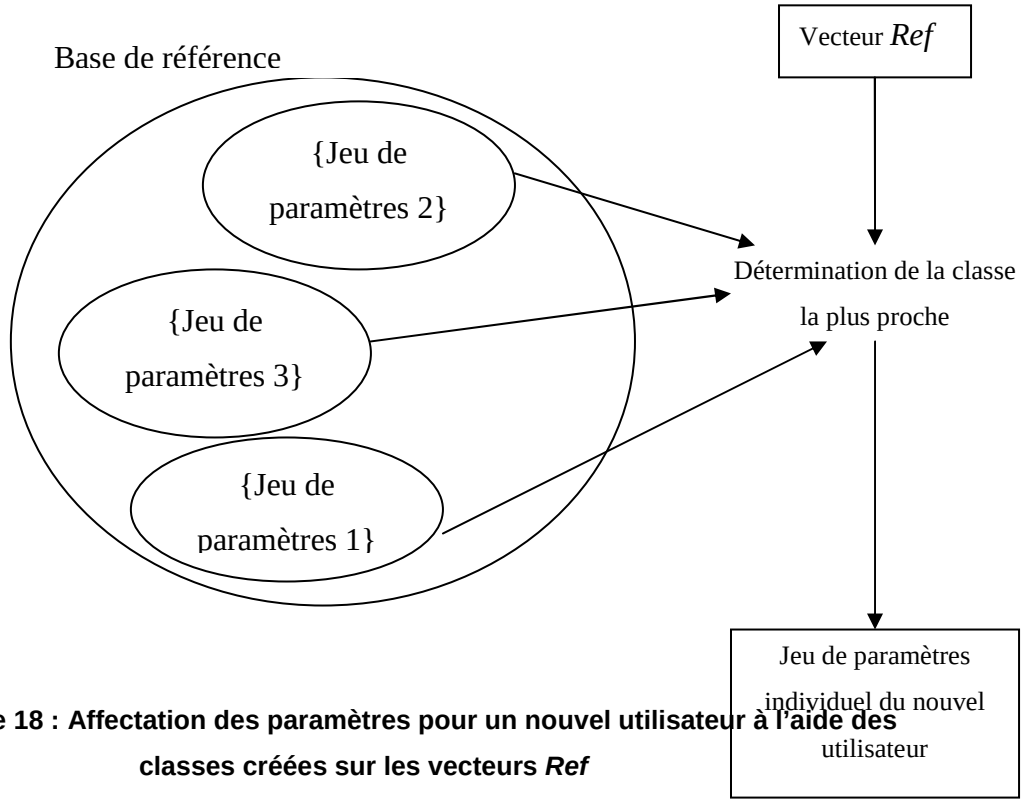


Figure 18 : Affectation des paramètres pour un nouvel utilisateur à l'aide des classes créées sur les vecteurs *Ref*

Pour déterminer les paramètres à associer au profil d'un nouvel utilisateur représenté par son vecteur *Ref* (Figure 18), nous calculons la distance entre ce vecteur et les vecteurs correspondant aux centres des classes. La classe associée au profil est celle dont le centre est situé à une distance minimum. Le jeu de paramètres propres à la classe obtenue est associé au nouvel individu.

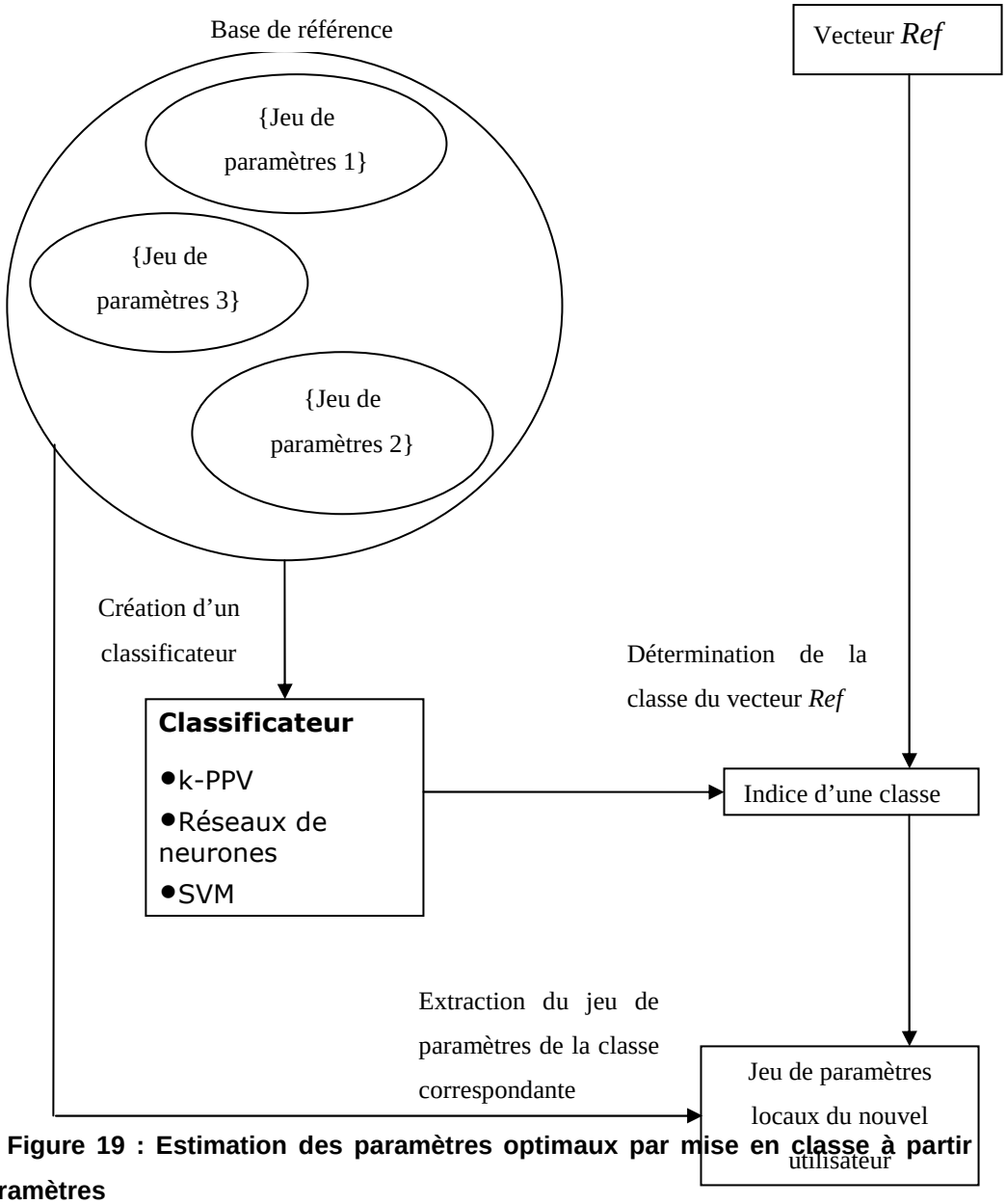


Figure 19 : Estimation des paramètres optimaux par mise en classe à partir des paramètres

2. La seconde solution que nous proposons (Figure 19) consiste à créer les classes d'individus à partir des valeurs des paramètres optimaux calculés sur les profils de la base de référence. Les regroupements ne sont plus réalisés à partir du comportement observé des utilisateurs lors de l'enregistrement mais à partir caractéristiques des individus présents dans la base de référence. Les paramètres individuels sont alors considérés comme des vecteurs de caractéristiques. Nous utilisons encore une fois les k-means pour créer des classes dans l'espace des paramètres. Après la phase de création des classes, nous disposons pour chacune d'un indice de classe et des coordonnées de son centre de gravité qui défini le jeu de paramètres à associer à chaque classe. Pour affecter un jeu de paramètres à un

nouveau profil, il est nécessaire de créer un classificateur dont l'ensemble d'apprentissage est constitué des vecteurs *Ref* de la base de référence. Quand un nouvel utilisateur s'enregistre dans le système un vecteur *Ref* est extrait de son profil et est présenté au classificateur qui lui affecte une classe et donc un jeu de paramètres. Divers classificateurs peuvent être utilisés pour réaliser cette opération. Dans la troisième partie du manuscrit, nous comparons les performances obtenues selon le type de classificateurs utilisés (SVM, réseau de neurones, k-ppv) dans le cadre d'applications concrètes.

Nous avons proposé dans cette partie plusieurs méthodes pour produire des paramètres individualisés pour chaque utilisateur ou groupes d'utilisateurs. Ces méthodes doivent être instanciées ou adaptées en fonction des applications biométriques à concevoir. Des exemples de mises en œuvre sont présentés dans les chapitres suivants. Ces exemples et expérimentations montrent à la fois l'intérêt et les difficultés associées à ces propositions. Leurs utilisations restent parfois problématiques quand l'ensemble d'apprentissage, c'est-à-dire la base de référence, est de petite taille. Néanmoins, ces méthodes améliorent les performances de la plupart des systèmes biométriques par rapport à l'utilisation de paramètres globaux.

2.4.2. Test de consistance des profils

Une fois le profil d'un utilisateur créé et les paramètres associés estimés, nous proposons d'inclure une étape de validation du profil. L'objectif de cette étape est de vérifier si le profil de l'utilisateur assurera un bon fonctionnement du système. Le nouveau profil ne doit pas engendrer des modifications telles que la réponse du système deviendra constante.

Ce problème se pose surtout dans la biométrie comportementale car, dans ce cas, l'utilisateur peut être peu expérimenté au départ et peut hésiter ou être stressé. Une mauvaise acquisition peut altérer un profil au point de le rendre inutilisable par la suite.

Cette phase, trouve aussi sa place dans un système basé sur une caractéristique issue de la biométrie physique. L'utilisateur peut, en effet, avoir des problèmes pour se placer ou bien n'avoir pas compris le fonctionnement du système. Il est alors utile de détecter les profils susceptibles de poser problème pour les éliminer.

Les profils inconsistants peuvent être détectés à l'aide d'une architecture à plusieurs niveaux. Dans un premier temps, les profils présentant d'importantes perturbations lors de l'acquisition peuvent être écartés. Pour cela, il est possible d'utiliser des méthodes empiriques comme le calcul de la distance maximale entre deux observations d'un profil, ou des heuristiques adaptées à la donnée biométrique utilisée (le nombre de temps anormalement longs dans la dynamique de frappe, la présence de longues pauses pour la reconnaissance de signature manuscrite en ligne...). L'utilisation de ce type d'heuristiques permet assez facilement de détecter la plupart des éléments rendant un profil inconsistant mais peine à détecter des modifications des comportements.

Dans ce dernier cas, nous proposons d'avoir recours à une analyse des valeurs prises par les paramètres personnalisés déterminées à l'aide des méthodes présentées précédemment. L'hypothèse que nous formulons, est que les paramètres d'un profil inconsistant se différencient des autres car ils présentent des valeurs extrêmes, principalement en ce qui concerne le seuil de sécurité. Il peut donc être opportun, après la phase d'estimation des paramètres, de repérer les profils avec des valeurs extrêmes et de les marquer comme inconsistants.

Nous proposons ensuite plusieurs solutions pour traiter les profils inconsistants. La première consiste à demander à l'utilisateur de recommencer la phase d'enregistrement. Ce n'est possible que si cette phase est assez courte. Dans le cas contraire, l'utilisateur déjà fatigué par son premier passage risque de produire des données non exploitables. Il est alors préférable de reporter la phase d'enregistrement à une date ultérieure.

La deuxième solution est d'utiliser une mise à jour du profil pour obtenir un profil consistant à moyen terme. Dans ce cas, il faut assouplir le seuil de décision le temps que les profils se stabilisent et deviennent consistants. La mise à jour du profil doit alors permettre d'actualiser le profil après chaque authentification réussie. Avec cette méthode, durant un laps de temps (qui peut être assez long), la sécurité du système risque d'être moindre pour certains utilisateurs. L'autre danger est que certains utilisateurs exploitent cette fonctionnalité volontairement lors de l'enregistrement et fournissent un profil inconsistant pour permettre à un tiers d'accéder au système. À notre avis cette dernière solution reste néanmoins la meilleure, car moins contraignante pour les utilisateurs. De plus, en cas de complicité interne, la sécurité d'un système est de toute façon très difficile à maintenir.

2.4.3. Mise à jour du profil

Le profil d'un individu est créé lors de la phase d'enregistrement à partir des données biométriques acquises. Le profil représente donc l'utilisateur à un instant précis. Un problème se pose alors lorsque, les caractéristiques biométriques de l'utilisateur évoluent au cours du temps. Pour les méthodes basées sur la biométrie physique, en règle générale, cette évolution est lente voire très lente. Il est donc possible de la négliger, pour peu que le processus d'acquisition ne soit pas influencé par l'habitude prise par l'utilisateur. Dans le cas contraire, la méthode aura une composante comportementale et doit donc être traitée comme telle.

La biométrie comportementale est sujette au problème de l'évolution des profils car les comportements utilisés pour l'authentification changent obligatoirement à cause de l'entraînement et de la répétitivité du processus d'acquisition (qui peut même devenir réflexe). Pour que le système fonctionne correctement dans la durée, il faut donc mettre à jour le profil à chaque nouvelle authentification réussie.

L'évolution du comportement d'un utilisateur comporte deux phases :

- La première phase correspond à l'apprentissage durant laquelle l'utilisateur se familiarise avec le dispositif et met au point son comportement. Selon la méthode choisie, cette phase a une durée variable, de quelques dizaines d'acquisition, à plusieurs mois d'utilisation du système. Durant cette première phase le profil de l'utilisateur évolue rapidement et les mises à jour doivent impérativement être régulières.
- La phase suivante correspond ensuite à une phase d'évolution naturelle. Elle commence lorsque que l'utilisateur maîtrise complètement le comportement étudié. Au cours de cette phase, le profil de l'utilisateur varie lentement. Des variations locales peuvent néanmoins être constatées à cause des modifications de comportement de l'utilisateur (fatigue, énervement…). Mais l'évolution reste globalement lente et le profil se stabilise. Nous pouvons donc a priori espacer les mises à jour durant cette phase. Il peut cependant apparaître des variations brutales du profil. Ces variations sont le plus souvent dues à un changement de comportement de l'utilisateur, ou à un changement de technique.

Notons également que la fréquence des mises à jour peut dépendre seulement de la vitesse d'évolution du profil mais également du coût des mises à jour.

Une fois la décision de mise à jour du profil prise, il existe plusieurs solutions pour l'ajustement du système :

- Réentraînement complet du système avec les dernières observations de l'utilisateur
- Rajout de la dernière observation aux précédentes en tentant, si possible, de mettre à jour le profil sans ré-entraîner les classificateurs
- Réacquisition du profil complètement

Chacune de ces méthodes a ses inconvénients et ses avantages. Si le moteur d'authentification est complètement ré-entraîné avec les dernières observations de l'utilisateur, le profil suit bien l'évolution du comportement au cours du temps mais avec une augmentation de la durée de reconnaissance qui peut être importante si l'entraînement est coûteux en temps machine. L'ensemble d'apprentissage reste alors de même taille, il est donc impossible, dans ce cas, de compenser un petit ensemble d'apprentissage.

Rajouter la nouvelle observation aux précédentes sans réentraîner tout le moteur peut sembler la solution idéale, mais cela n'est possible que si les classificateurs le permettent. Un mécanisme d'oubli doit être également implémenté afin que le profil continue à évoluer à très long terme. Ce mécanisme peut par exemple être un mécanisme de poids, attribués aux observations : les plus récentes ayant un poids plus grand que les anciennes. Cette méthode est la plus avantageuse. En effet, il est ainsi possible d'agrandir l'ensemble d'apprentissage en y incorporant les dernières observations. Un système de poids évolutif peut également être mis en place en cas de variation brusque du profil pour mieux suivre son évolution.

Réacquérir entièrement le profil, n'est intéressant que si une trop grande variation est constatée ou si l'utilisateur a du mal à s'authentifier.

Pour conclure cette partie nous estimons la mise à jour du profil indispensable pour toutes les données biométriques sujettes à une évolution non négligeable. La méthode que nous préconisons est de mettre à jour le profil en ajoutant la dernière observation acceptée à l'ensemble d'apprentissage, même si pour certains classificateurs, la phase d'entraînement doit alors être entièrement reconduite. La perte de temps engendrée est préférable à un système ne fonctionnant pas correctement.

2.5. Bilan : La base de référence est un élément clé

Dans ce chapitre, nous avons présenté et préconisé différentes techniques utiles dans le cadre de la mise en place d'un système d'authentification biométrique comportementale. Notre objectif était d'identifier les points critiques de ces systèmes et d'apporter des propositions de solutions pouvant être appliquées quel que soit le type de données choisies.

Parmi ces problèmes, deux ressortent particulièrement : la variabilité du profil d'un utilisateur au cours du temps que nous proposons de résoudre à l'aide de l'implémentation de méthodes de mises jour régulières des profils et la variabilité des comportements des utilisateurs partiellement solutionnée par la mise en œuvre de techniques de personnalisation du système pour tous les utilisateurs.

Selon nous, une solution simple et efficace pour réaliser la personnalisation est d'intégrer une phase de fusion qui privilégie les classificateurs qui fonctionnent le mieux selon les comportements des utilisateurs. Les paramètres des différents classificateurs, leur poids et le seuil de sécurité sont déterminés par différentes méthodes que nous avons détaillées dans ce chapitre. Le point commun de chacune de ces méthodes est l'utilisation d'une base de référence.

La qualité de la base de référence influence donc grandement les performances des systèmes tels que nous les proposons. Il est donc indispensable de réfléchir sur ce qu'est une bonne base de référence. Nous pouvons citer les conditions suivantes :

- Diversité de la population : il est impératif d'essayer d'étendre la base de référence sur un maximum d'utilisateurs et de ne pas se limiter à une population donnée (étudiant par exemple).
- Stabilité du contexte l'acquisition : Le contexte d'acquisition doit respecter des conditions clairement définie (le matériel utilisé doit être similaire…)
- Stabilité de la procédure d'acquisition : les utilisateurs doivent suivre, lors de l'acquisition de la base de référence, la procédure qui sera retenue au final pour l'enregistrement et les autres acquisitions.

Les recommandations que nous avons présentées pour chaque partie du système ont pour but d'aider les concepteurs à réaliser une architecture cohérente et adaptée. Nous préconisons également une procédure d'évaluation des systèmes intégrant des critères autres que les simples taux de reconnaissance habituellement diffusés. Le prochain chapitre illustre comment il est possible de mettre en place et d'évaluer, en suivant nos recommandations et propositions, des applications concrètes. Ces expérimentations permettent notamment d'évaluer l'influence de nos propositions sur les performances d'un système biométrique.

Chapitre 3.
Application à l'analyse de la dynamique de frappe

Dans ce chapitre, nous proposons une solution à un problème de sécurité concernant l'accès à des ressources informatiques. L'élaboration de cette solution utilise l'architecture et les recommandations que nous avons présentées dans le chapitre précédent. Nous commençons donc par présenter les contraintes auxquelles est soumis le système. Ces contraintes contraindront le choix de la méthode biométrique utilisée ainsi que son mode de mise en place. Ensuite, nous présentons les résultats de son évaluation en pointant les améliorations apportées par notre architecture.

3.1. Choix des caractéristiques et phase d'enregistrement

3.1.1. Contraintes et méthodes biométriques choisies

Le problème d'authentification biométrique à résoudre nous a été proposé par la Société CAPMONETIQUE qui a financé cette thèse. Cette société souhaite mettre en place un système d'authentification léger et peu cher basé sur la biométrie. Le problème à résoudre est le contrôle d'accès à des ressources informatiques. Le constat de départ est que le traditionnel couple identifiant/mot de passe propose une sécurité insuffisante pour les applications actuelles. En effet des études, [Thompson, 1979] et [Yan *et al.*, 2004] par exemple, ont montré qu'il était sujet à de très nombreux problèmes et ce depuis des lustres. On peut notamment citer :

- la possibilité de les trouver par « force brute » c'est-à-dire en essayant tous les mots de passe possibles à l'aide de dictionnaires (en moins de cinq minutes plus de 75% des mots de passe sont trouvés)
- le syndrome du post-it : quand les administrateurs, dans le but de renforcer la sécurité, imposent de longs mots de passe ou des changements fréquents, beaucoup d'utilisateurs écrivent leurs mots de passe sur un post-it et le colle sous le clavier voire même sur l'écran de leur ordinateur

- L'utilisation de *key-loggers* : ce sont des logiciels de type spyware qui interceptent les frappes au clavier d'un utilisateur et permettent donc d'enregistrer le mot de passe
- La complicité d'un utilisateur : un utilisateur du système peut divulguer intentionnellement ou non son mot de passe à une tierce personne
- Ou plus simplement, un utilisateur peut regarder « par dessus l'épaule » d'un autre utilisateur, pour découvrir son mot de passe

L'objectif de notre étude est donc d'essayer de résoudre ces problèmes, notamment en empêchant le prêt et le vol d'identifiant. Une contrainte essentielle que devra respecter le système retenu est qu'il doit imposer le minimum de contraintes pour les utilisateurs. Une autre volonté importante est de limiter l'ajout de nouveau matériel afin de construire une solution bon marché.

Ces contraintes suggèrent naturellement l'utilisation de méthodes issues de la biométrie comportementale. Pour contrôler l'utilisation de ressources informatiques sans coût matériel supplémentaire, nous avons à notre disposition deux périphériques présents sur tous les ordinateurs : la souris et le clavier. Les avantages des méthodes basées sur ces deux périphériques sont :

- Leurs coûts très faibles : puisque aucun matériel ne doit être rajouté
- Leur facilité d'utilisation : tout le monde a déjà utilisé un clavier et une souris
- La bonne acceptation de ce type de méthodes par les utilisateurs : les données acquises ne sont pas considérées comme cruciales pour la vie privée

Le seul inconvénient de ces méthodes est leurs performances qui sont très en dessous de celles des méthodes biométriques basées sur des caractéristiques physiques. Mais, comme les contraintes de sécurité du système ne sont pas critiques, une amélioration significative des performances pourraient suffire pour notre application.

Nous avions envisagés de coupler la dynamique de frappe à l'étude de la manipulation de la souris, mais des tests préliminaires nous ont montré que l'utilisation de la souris pour l'authentification pose problème. En effet, les influences du type, de la sensibilité et de la propreté de la souris sont très importantes

et donc trop perturbantes pour la reconnaissance. Néanmoins, avec la généralisation des souris optiques, et la possibilité de régler leurs résolutions, il pourrait être intéressant de refaire ces tests.

Nous avons donc choisi de nous limiter à l'utilisation de la dynamique de frappe.

3.1.2. Choix de la séquence de reconnaissance

Une fois la caractéristique choisie : la dynamique de frappe, une réflexion a dû être menée sur la manière d'utiliser cette caractéristique biométrique.

L'authentification de l'utilisateur par analyse de sa dynamique de frappe peut se faire de deux façons distinctes :

- A l'aide d'une séquence de login saisie, lors de l'ouverture de la session par exemple. Dans ce cas, l'authentification est réalisée en une fois et si l'utilisateur échoue la session ne s'ouvre pas. Le problème est que si celui-ci quitte son poste de travail sans verrouiller la session n'importe qui peut le remplacer et en l'absence de contrôle faire ce qu'il veut.
- Par monitoring continu de l'utilisateur lors de son travail au clavier. Le système examine, à intervalles réguliers, la correspondance entre la dynamique de l'utilisateur actuel et celle stockée dans le profil courant. L'inconvénient de la surveillance continue est qu'un imposteur peut accéder à beaucoup d'informations et faire beaucoup de dégâts avec peu ou pas d'interaction avec le clavier, en utilisant uniquement la souris par exemple. Une application pourrait cependant être la surveillance d'examens en ligne, puisque de grandes quantités de textes doivent être saisies, et il est donc possible de détecter d'éventuelles substitutions d'utilisateur.

De nos jours, le clavier n'est plus indispensable pour réaliser la plupart des tâches sous un système d'exploitation avec interface graphique comme Windows. Nous pensons donc que la surveillance continue d'un utilisateur n'est pas suffisante. Nous avons donc décidé de nous concentrer, dans un premier temps, sur la reconnaissance limitée à la séquence de login.

Le choix suivant a concerné le type de la séquence à analyser. En fait, l'analyse de la dynamique de frappe peut se faire sur deux types séquences :

- Les séquences libres : les séquences de touches à saisir ne sont pas fixées a priori. Elles peuvent même être différentes lors de chaque demande

d'authentification d'un utilisateur. Ces séquences peuvent de plus être laissées au choix de l'utilisateur ou bien être générées par le système.

- Les séquences fixes : la séquence à saisir par un utilisateur est déterminée une fois pour toutes lors de l'enregistrement et reste toujours identique. C'est le cas par exemple lors de l'utilisation de mots de passe.

L'avantage des séquences libres est de ne pas imposer à l'utilisateur de mémoriser une séquence. Il est donc authentifié uniquement sur sa dynamique de frappe. Dans le cas des séquences fixes, l'utilisateur est authentifié grâce à la séquence et à sa dynamique de frappe ce qui complique la tâche des imposteurs qui doivent déterminer la séquence avant d'essayer de reproduire la dynamique de frappe elle-même. Par contre, cette méthode impose une gestion du stockage des mots de passe.

Le but du système désiré par Capmonétique est de faire mieux que les mots de passe traditionnels, nous jugeons donc que nous pouvons les réutiliser et ajouter un contrôle de la dynamique de frappe afin de renforcer la sécurité.

Avant de présenter les choix suivants, qui sont fortement dépendant du choix de la méthode biométrique choisie, il nous a semblé important de faire un bref récapitulatif des travaux existants traitant de la dynamique de frappe.

3.1.3. La dynamique de frappe

La dynamique de frappe est l'étude de caractéristiques biométriques extraites à partir d'une analyse de la manière dont un utilisateur utilise un périphérique à touches (clavier d'ordinateur, téléphone portable…) afin de différencier des individus.

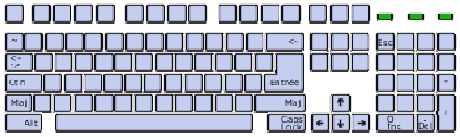
L'analyse de la dynamique de frappe est apparue bien avant les ordinateurs et les claviers. En effet, dès l'époque du télégraphe, les opérateurs étaient capables de se reconnaître simplement grâce au rythme avec lequel ils envoyaient les impulsions en Morse. Mais, il a fallu attendre les années 1980 pour voir apparaître les premiers travaux scientifiques démontrant qu'il était possible de différencier des individus à partir leur façon de taper aux claviers grâce à des tests statistiques simples [Gaines *et al.,* 1980]. Le développement de la dynamique de frappe s'est considérablement accéléré ces dernières années, notamment avec la mise en place d'un système de reconnaissance commercial [Biopassword].

Le clavier est aujourd'hui l'un des deux principaux périphériques d'entrée utilisés pour communiquer avec son ordinateur. Il est quasiment présent sur la totalité des ordinateurs. Mais, il existe sous divers formats et apparences (Figure 20) qui entraînent de grandes différences dans la dynamique de frappe de chaque utilisateur.

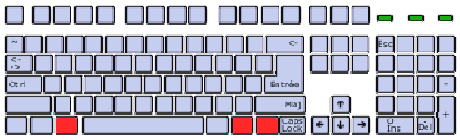
Les claviers de portables sont aujourd'hui des variations des claviers compatibles Windows mais avec une touche fonction supplémentaire dont l'emplacement varie suivant le constructeur.

Les claviers de type PC/XT

Les claviers de type PC/AT



Les claviers étendus

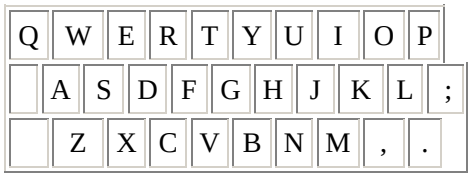


Les claviers compatibles Windows

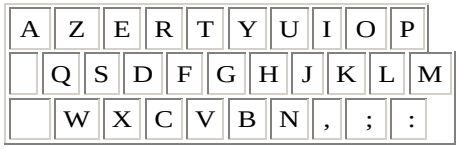
Figure 20 : Les différents types de claviers

En plus de ces différents types de clavier, il existe une différence entre les dispositions des claviers de différents pays (Figure 21) ; en France les claviers sont de type AZERTY alors que le modèle américain est de type QWERTY. Pour expliquer la différence de disposition des touches, il faut revenir à la machine à écrire. A l'époque, pour éviter que les tiges portant les caractères ne se croisent et se bloquent entre elles, les caractères se suivant le plus souvent dans la langue ont été éloignés les uns des autres. Par la suite, un projet de clavier « optimisé » a vu le jour, proposé par Dvorak, ce clavier était sensé optimiser la vitesse de frappe. Mais son

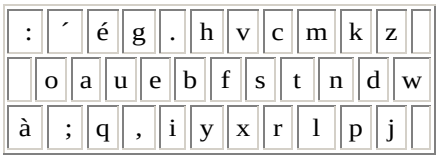
gain de performance limité par rapport aux coûts de migration a entraîné son abandon.



Clavier QWERTY



Clavier AZERTY



Clavier « idéal » Dvorak en français

Figure 21 : Dispositions des touches suivant les pays

En plus de ce changement de disposition, on assiste à l'apparition de clavier dit « ergonomique ». Comme par exemple le Microsoft Natural Keyboard Elite (Figure 22). Ces claviers ont la même disposition de touches que les claviers classiques mais possèdent une forme différente visant à faciliter la frappe.

Figure 22 : Microsoft Natural Keyboard Elite

Il est probable que suivant la configuration des touches, et suivant le style de clavier le profil de dynamique de frappe évolue lors d'un changement de clavier entraînant ainsi une difficulté supplémentaire.

En plus de la diversité de ces claviers pour ordinateur, il est possible d'appliquer la dynamique de frappe sur des claviers de téléphones ou des digicodes (voir [Ord et Furnell, 2000] par exemple) même si les performances sont sensiblement altérées par rapport à l'analyse de la dynamique de frappe sur claviers classiques.

3.1.3.1. Gestion du clavier sous Windows

Notre étude devant aboutir à la création d'un logiciel sous Windows, il est intéressant de voir comment ce système d'exploitation gère ce périphérique. A chaque pression d'une touche, le clavier envoie un message au système d'exploitation contenant le code de la touche pressée. Le système d'exploitation décide ensuite de l'action à effectuer. En règle générale, le système lève une interruption ou son équivalent sous Windows (les *hooks*) et la transmet à l'application chargée de la traiter. Pour récupérer les informations clavier, on peut ainsi, soit intercepter l'événement clavier avant que Windows ne le traite, par exemple en réécrivant un *hook*, soit créer une application qui sera destinataire des événements clavier.

3.1.3.2. Les caractéristiques extraites

Dans le cadre de la dynamique de frappe, les seules informations généralement disponibles sont les dates auxquelles ont lieu des événements clavier (pression ou relâchement d'une touche), l'ordre d'appuis et de relâchements des touches (données non numériques) et d'autres caractéristiques pouvant être extraites par l'intermédiaire de claviers spéciaux (notamment la force de la pression exercée sur les touches lors de la frappe). Dans notre étude, nous nous intéressons uniquement aux caractéristiques disponibles classiquement afin de ne pas perdre l'un des grands avantages de la dynamique de frappe, l'absence de coût d'achat de matériel supplémentaire.

Nous avons déjà mentionné les deux méthodes possibles de construction du vecteur de caractéristiques dans le cadre de l'utilisation du clavier pour l'identification ou l'authentification : la séquence à taper peut être fixée une fois pour

toutes lors de la phase d'enregistrement (login/mot de passe) ou bien entièrement libre (surveillance continue d'un utilisateur au cours de son travail habituel). Les vecteurs de caractéristiques ne sont pas identiques dans ces deux cas ; néanmoins, quelle que soit la méthode choisie, les mêmes informations temporelles sont extraites en regardant un couple de touches successivement enfoncées. Plusieurs types de temps peuvent être étudiés lors de la frappe d'un couple de touches (Figure 23) :

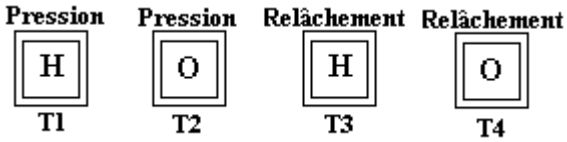


Figure 23 : Temps extraits au cours de la frappe des touches H et O

En ne s'intéressant qu'à des séries de deux touches successives les temps suivants peuvent être construits :

- P-P (Press-Press) : temps entre deux pressions de touches (T2-T1)
- P-R (Press-Release) : temps entre l'appui sur la touche et le moment où elle est relâchée (T3-T1 et T4-T2)
- R-P (Release-Press) : temps entre la relâche d'une touche et l'appui sur la suivante (T3-T2)
- R-R (Release-Release) : temps entre la relâche de deux touches successives (T4-T3)

Il est possible d'étudier les temps associés à plus de deux touches successives, mais aucune recherche n'a, à notre connaissance, été menée sur cette possibilité.

Ensuite, la construction des vecteurs finaux dépend de la méthode d'acquisition choisie.

- Cas d'une séquence libre : Dans ce cas tous les couples de touches successives susceptibles d'apparaître dans un texte sont considérés. Pour chacun de ces couples, les temps PP, PR, RP et RR sont calculés et le vecteur de caractéristiques contient alors pour chaque couple de touches, les codes des touches elles-mêmes et les quatre temps décris dans la section précédente. En plus de ces temps classiques, il est possible d'examiner d'autres caractéristiques : On peut notamment examiner le nombre d'erreurs au cours de la frappe, indiquée par la fréquence d'utilisation des touches « Suppr » et « backspace », ou encore des données non numériques comme l'ordre de relâchement des touches lors de

l'utilisation des touches « majuscules ». Pour l'instant ces données n'ont jamais été utilisées dans la littérature.

- Cas d'une séquence fixe : Pour extraire les caractéristiques lors de la frappe d'une séquence imposée, la première possibilité est de traiter la séquence comme s'il s'agissait d'une séquence libre. Dans ce cas, on considère que deux couples identiques placés à deux endroits différents de la séquence, participe à la génération d'une seule caractéristique au lieu de deux différentes. Cela entraîne donc une perte d'information. L'autre possibilité est de prendre en compte uniquement les touches présentes dans la séquence en les considérants par ordre d'apparition. Les temps classiques sont extraits pour chaque couple de touches. Ces temps mis bout à bout constituent le vecteur de caractéristiques dont la longueur dépend donc de la longueur de la séquences utilisée.

3.1.3.3. Méthode d'analyse

Même si les premières études portant sur la dynamique de frappe datent de 1980 [Gaines *et al.*, 1980], la dynamique de frappe n'a réellement commencé a se développer que depuis les années 2000. Plusieurs grandes tendances se sont alors développées :

- Les premières méthodes utilisaient des méthodes statistiques simples qui permettent d'obtenir de bonnes performances lorsque les ensembles d'apprentissage sont de faible taille.
- Parallèlement, d'autres auteurs ont utilisé des réseaux de neurones dans des problèmes d'identifications obtenant également des résultats corrects.
- Les dernières tendances utilisent des classificateurs plus sophistiqués comme les SVM à une classe, les SVDD… Le problème de ces méthodes est l'obligation d'utiliser des ensembles d'apprentissage de très grande taille (plus de 50 séquences).

Tableau 2 : Résumé des résultats obtenus par les méthodes actuelles

Problème à une classe

Référence	Méthode	Taille de la séquence de reconnaissance en caractères	Taille de la séquence d'enregistrement	Taille de la base de test	Performance
[Kacholia et Pandit, 2003]	Distribution de Cauchy	12	11 séquences de reconnaissance	20	TFA : 1 % TFR : 4,8%
[Monrose et Rubin, 2000]	Mesure de similarité	Texte libre	Texte libre	63	TBC=92,4%
[Yu et Cho, 2004]	SVM à une classe	6-10	50 séquences de reconnaissance	21	TFA : 0 % TFR : 6,28%
[Guven et Sogukpinar, 2003]	Mesure de similarité par des cosinus	8	1 séquence de reconnaissance	12	TFA : 1 % TFR : 11,7%
[Bergadano *et al.*, 2002]	Mesure du désordre	texte libre de 683 caractères	1 séquence de reconnaissance	44	TFA : 0,04 % TFR : 4%
[Leggett *et al.*, 1991]	Méthode statistique	500 Texte libre	1000 Texte libre	17	TFA : 5,5 % TFR : 5%
[Coltell *et al.*, 1999]	Méthode statistique	20	20 séquences de reconnaissance	10	TFA : 5 % TFR : 30%
[Obaidat et Sadoun, 1997]	Réseau de neurones : ART2/BPN/ CPN	30	20 séquences de reconnaissance	6	TBC =97,5 % TCB=95,8 % TCB=89,17%
[Anagun et Cin, 1998]	Réseau de neurone : BPN	7-14	?	5	TBC=83,7 %TFA=8,2%
[Ord et Furnell, 2000]	Réseau de neurone BPN	6 chiffres	30 séquences de reconnaissance	14	TFA : 9 % TFR : 30%
[Lee et Cho, 2005]	SVDD-1-LVQ	6-10	76 séquences de reconnaissance	21	Erreur intégrée : 0,43/0,62
[Sheng *et al.*, 2005]	Arbre de décision	37	9 séquences de reconnaissance	43	TFA : 0,8 % TFR : 9,6%
[Chen et Chang, 2004]	HMM	login	20 séquences de reconnaissance	20	TEE=1,2%
[Napier *et al.*, 1995]	Khi 2	300 Texte libre	1 séquence de reconnaissance	24	TEE=10%
[Furnell *et al.*, 1995]	M/V	160 Texte libre	2*2200 texte libre	26	TFA : 0 % TFR : 15%
[Bleha *et al.*, 1990]	Model statistique	20	10 séquences de reconnaissance	32	TFA : 2,8 % TFR : 8,1%
[Cho *et al.*, 2000]	Neural network	7	75 séquences de reconnaissance	21	TFA : 0 % TFR : 1%

Le Tableau 2 indique les résultats obtenus par la plupart des méthodes de la littérature scientifique basées sur la dynamique de frappe. La première remarque est la grande disparité des résultats entre les méthodes. Ces différences s'expliquent par la taille des bases de test étudiées extrêmement variable allant de 5 [Anagun et Cin, 1998] à 63 [Monrose et Rubin, 2000] utilisateurs. On remarque que même la plus grande des bases de test ne comporte que 63 utilisateurs ! Ce manque de grandes bases de données publiques entraîne un manque de lisibilité des résultats. Une autre explication des grands écarts obtenus provient de la différence dans les protocoles de tests utilisés : tailles différentes pour l'ensemble d'apprentissage, longueur de la séquence, durée du test. De même le seuil de sécurité choisi est différent pour chaque test, difficile de comparer une méthode ayant imposé 0 en TFA avec une méthode choisissant de présenter le TEE. L'ensemble de ces éléments rend difficile les comparaisons entre méthodes.

Néanmoins, le premier enseignement pouvant être tiré de ce tableau, est que les méthodes utilisées dans le but de résoudre un problème à une classe et celles pour un problème à deux classes sont très différentes. Dans le cas du problème à deux classes, que ce soit pour l'authentification ou l'identification, les méthodes utilisées sont les réseaux de neurones ([Obaidat et Sadoun, 1997] [Anagun et Cin, 1998] [Ord et Furnell, 2000]…) ou d'autres classificateurs usuels [Sheng *et al.*, 2005]. Dans le cadre du problème à une classe, les méthodes les plus populaires sont des classificateurs statistiques basés sur les moyennes et les variances [Leggett *et al.*, 1991], [Coltell *et al.*, 1999] et des mesures de similarité [Guven et Sogukpinar, 2003], [Bergadano *et al.*, 2002]. Des classificateurs plus sophistiqués comme les SVDD [Lee et Cho, 2005], les réseaux de neurones auto-associatifs [Cho *et al.*, 2000], et les SVM à une classe [Yu et Cho, 2004] sont également utilisés dans le cadre du problème à une classe (grisé sur le tableau).

Un autre enseignement apporté par cet état de l'art est l'importance de la sélection de caractéristiques prouvée notamment dans [Yu et Cho, 2004]. Cette sélection de caractéristiques, réalisée à l'aide d'un algorithme génétique, permet d'accroître considérablement les performances des SVM à une classe mais au prix d'une grande complexification du système. Il semble alors indispensable d'utiliser un ensemble d'apprentissage de grande taille.

Pour notre problématique, il faudra veiller à choisir des classificateurs respectant les contraintes d'utilisabilité et de faibles coûts de mises en place exprimées par CapMonétique.

3.1.4. Choix des séquences pour la reconnaissance

Le choix des séquences pour l'authentification a été fait avec la volonté d'imposer un minimum de contraintes aux utilisateurs.

Pour ce faire et pour ne pas désorienter les utilisateurs, nous avons décidé de reconnaître un utilisateur uniquement à l'aide du couple identifiant/ mot de passe.

Chaque utilisateur choisit un identifiant et un mot de passe différent. Ce choix ne permet pas d'utiliser des données d'imposteurs pour créer les profils. On se trouve donc dans le cadre de la résolution d'un problème à une classe.

Le nombre de séquences pouvant être demandées à un utilisateur constitue un choix difficile. En effet, saisir plusieurs séquences peut être très fatiguant pour un utilisateur. Ce dernier peut se lasser, devenir déconcentré et son comportement peut donc changer. D'un autre côté, il est nécessaire d'acquérir suffisamment de séquences pour créer un profil représentatif de son comportement au clavier.

En étudiant les comportements d'utilisateurs, au moment de l'enregistrement, nos tests ont montré que le juste équilibre se trouvait aux alentours de la saisie d'une dizaine de séquences (comportant de 8 à 20 caractères). Au-delà, les erreurs de saisie deviennent fréquentes, liées à une déconcentration, à la lassitude ou à l'énervement. Nous avons donc décidé de nous limiter à 10 séquences pour construire les profils des utilisateurs.

3.1.5. Mise en place de la base de référence

Nous avons vu dans le chapitre précédent l'intérêt de disposer d'une base de référence pour construire le système. Pour notre application, nous disposons de données fournies par 48 utilisateurs sur une durée variant de quelques semaines à 3 ans. Ces informations ont été, pour la plupart, acquises sans contrôle de notre part. Un logiciel d'acquisition lancé automatiquement au démarrage de l'ordinateur ou sur intervention de l'utilisateur a permis l'acquisition des séquences de login. Nous avons dû rejeter huit utilisateurs à cause du trop faible nombre de séquences fournies

(uniquement les dix séquences d'apprentissage), ou à cause d'une mauvaise manipulation des utilisateurs (suppression du répertoire de stockage des données, prêt de machine à une tierce personne,…).

Notre base comprend donc finalement 40 utilisateurs, issus de plusieurs milieux : étudiants et enseignants d'une école d'ingénieur en informatique, employés de divers services d'une société, personnes appartenant à divers cercles familiaux. La population est donc extrêmement diversifiée pour ce qui concerne la fréquence d'utilisation du clavier. Elle contient des personnes novices ou n'utilisant l'outil informatique que de façon très irrégulière, des étudiants ayant un bon niveau mais sans formation spécifique autre qu'une pratique courante, des employés utilisant l'informatique dans leur travail mais de façon sporadique et deux secrétaires ayant subit un entraînement spécifique à la dactylographie. Le matériel utilisé pour l'acquisition comprend des claviers classiques en grande majorité et trois utilisateurs ayant réalisé des acquisitions sur des ordinateurs portables. Cette grande variété implique une grande disparité de comportements dans notre base, ce qui nous permettra de bien évaluer l'influence de la personnalisation du système pour chaque utilisateur.

La longue durée d'acquisition, pour certains des utilisateurs (3 ans au maximum, et 10 utilisateurs ayant fournis des informations sur plus de 6 mois) nous permettra également de justifier l'intérêt de la mise à jour du profil.

Cette base se divise en deux parties. La première partie se compose de 20 utilisateurs ayant tous la même séquence de login. Pour cette partie de la base, il sera possible de disposer d'attaques d'imposteurs construites à partir des données des autres utilisateurs utilisant les mêmes séquences. Ces utilisateurs ont fournit entre 20 et 120 séquences. Il est donc possible de disposer, pour chaque utilisateur, de 600 attaques provenant des logins d'autres utilisateurs.

L'autre partie de la base est constituée d'utilisateurs ayant des identifiants et mots de passe différents. Ces utilisateurs ont fourni entre 30 et 220 séquences. Les attaques d'imposteurs ont été produites par des utilisateurs auxquels les mots de passe et identifiants ont été divulgués. Les attaques ont été enregistrées par séries de 10 séquences. Ainsi chaque utilisateur a été attaqué entre 30 et 150 fois pour cette partie de la base.

Ce découpage a pour conséquence de produire deux bases différentes (séquences identiques pour tous et séquences différentes) que nous considérons de

taille insuffisante pour créer deux bases complètement distinctes (avec pour chacune, une base de référence, une base de test et une base de validation). Pour nos tests, sauf mention contraire, les deux bases sont toujours utilisées conjointement afin de disposer d'un maximum d'échantillons.

De plus, toujours compte-tenu de la taille de notre base, il a été impossible de la découper pour produire des bases distinctes d'évaluation et de développement. Ainsi, malgré nos efforts, il est à noter que cette situation pourra peser légérement sur la fiabilité de nos résultats.

La meilleure solution nous a semblé être d'utiliser, lors de nos expérimentations, la procédure d'évaluation par validation croisée (*leave one out*) présentée dans les chapitres précédents. Cette procédure d'évaluation permet de confondre, au prix d'une baisse de fiabilité des mesures, la base de référence et la base de test. Lorsque les performances d'un utilisateur sont évaluées, la base de référence est constituée des données de tous les autres utilisateurs. Au final, pour tester chaque utilisateur, on dispose d'une base de référence comprenant les profils des 39 autres utilisateurs, les poids et les seuils sont calculés grâce à cette base.

Afin de mieux quantifier les biais introduits par ces choix, nous préciserons pour chacune de nos expérimentations les intervalles de confiance et significativité des résultats produits (pour les TEE'). Les intervalles de confiance sont calculés (de manière classique : FORMULE) avec une marge de 5% et donnent une indication sur la stabilité des résultats obtenus. Notons d'ores et déjà que les intervalles de confiances ont de grandes chances d'être non négligeables lorsqu'ils seront mis en rapport avec l'ordre de grandeur de nos résultats du fait du faible nombre d'utilisateurs dans la base et de la grande variabilité inter-utilisateurs.

3.1.6. Procédure d'évaluation

Pour évaluer notre système, nous utilisons les taux classiques comme le TFR et le TFA, ainsi que le taux TEE permettant de comparer plus facilement deux méthodes. Ce taux est hélas parfois difficile à obtenir ; c'est pourquoi nous utilisons alors une approximation, le TEE'que nous définissons comme la moyenne du TFA et du TFR. Ce taux sert à comparer facilement les différentes méthodes entre elles.

Les dix premières séquences de chaque utilisateur sont toujours réservées à la création des profils utilisateurs. Ensuite les taux d'erreur sont calculés, pour un utilisateur, de la façon suivante :

- Le TFR est calculé en testant, par ordre chronologique de saisie, toutes les séquences produites par cet utilisateur et en prenant le taux de séquence rejeté.
- Le TFA est calculé à l'aide des séquences des imposteurs. Si la mise à jour du profil est activée, nous re-testons après chaque login réussi (donc chaque mise à jour) tous les logins des imposteurs. Si la mise à jour n'est pas activée les séquences des imposteurs ne sont testées qu'une seule fois.

Les taux d'erreur utilisés pour mesurer les performances de notre système sont calculés à partir des TFR et TFA de tous les utilisateurs. Ces mesures sont le TFR moyen et le TFA moyen qui produisent une bonne estimation des performances du système. Il sera aussi possible de regarder les taux d'erreur maximum pour voir comment le système se comporte avec les utilisateurs à problème.

Dans le chapitre 2, nous indiquions qu'il est indispensable d'accompagner ces mesures classiques par d'autres indicateurs (taux d'erreur à l'enrôlement, coût du système, acceptation des utilisateurs. Nous présenterons ces divers indicateurs après avoir mis en place notre système.

3.2. Classificateurs et architecture du système de décision

Les contraintes imposées par l'application, ne permettent pas d'utiliser les classificateurs les plus compliqués (SVM à une classe, SVDD) principalement à cause de la faible taille de l'ensemble d'apprentissage. Nous avons donc choisi d'utiliser des classificateurs plus simples basés sur des mesures statistiques. Par contre, il est possible d'utiliser plusieurs classificateurs et donc de valider notre recommandation de mise en place d'une phase de fusion dans les systèmes d'authentifications biométriques. Les trois classificateurs choisis sont décrits dans les sections qui suivent.

3.2.1. Méthode basée sur les moyennes et les variances

Le premier classificateur est dérivé d'une méthode proposée pour authentifier des utilisateurs par Leggett et al dans [Leggett *et al.*, 1991] et [Leggett et Williams, 1988]. Pour la création du profil, la moyenne et la variance de chaque temps associées aux couples de touches successives présentes dans la séquence sont calculées. Ensuite, cette méthode utilise un test basé sur la moyenne μ et l'écart type σ de chaque caractéristique (temps). Un temps *o* est déclaré valide s'il est situé à moins de 0,5 écart type de la moyenne calculé sur l'ensemble d'apprentissage (équation (17)).

$$|o - \mu| < 0{,}5.\sigma \qquad \textbf{(17)}$$

Durant la reconnaissance, une séquence est considérée comme valide si 60% des temps extraits au cours de la frappe sont valides, ce qui correspond à la définition d'un seuil de sécurité. Cette méthode a ensuite été améliorée dans [Coltell *et al.*, 1999] par l'introduction d'un paramètre supplémentaire α (équation (18)).

$$|o - \mu| < \alpha.\sigma \qquad \textbf{(18)}$$

Cette méthode revient donc à définir une sorte de tunnel pour chaque caractéristique autour de la moyenne dont la largeur est définie à la fois par son écart type et par α.

La décision est prise ensuite en réalisant un seuillage sur le nombre de temps non valides. Nous avons décidé d'apporter des améliorations à cette méthode notamment dans le but d'éviter les effets de bord et de n'avoir plus qu'un paramètre à régler : un seuil unique. Un score est associé à chaque temps par l'équation (19) qui s'inspire du z-score.

$$score = e^{-\frac{|o - \mu|}{\sigma}} \qquad \textbf{(19)}$$

Le score global de l'observation est la moyenne des scores obtenus pour chaque caractéristique (temps). Ce score moyen pourra être comparé à un seuil pour prendre la décision finale.

Nous avons mesuré les performances de ce classificateur sur notre base afin d'évaluer ses performances et ses points faibles. Nous avons évalué les performances

de ce classificateur à l'aide de la partie de la base qui contient les utilisateurs ayant des mots de passe identiques. Nous avons fait ce choix pour s'affranchir, dans un premier temps, des difficultés liées à la personnalisation du seuil et à la normalisation des scores. Le seuil est identique pour tous les utilisateurs et réglé de manière à se rapprocher le plus possible du TEE.

Tableau 3 : Performances du classificateur basé sur des mesures statistiques

Méthode	TEE'	TFA moyen	TFR moyen	TFA max	TFR max	Intervalles de confiance à 95% Pour le TEE'
Méthode initiale [Coltell *et al.*, 1999]	4,6%	4,4%	4,8%	27,1%	13,0%	[2,6 ; 6,6]
Méthode améliorée [Hocquet *et al.*, 2005]	3,6%	3,6%	3,6%	21,6%	10,0%	[2,0 ; 5,2]

Les performances de ce classificateur sont présentées dans le Tableau 3. On observe que cette méthode a des performances voisines de celles obtenues dans les précédentes études [Leggett *et al.*, 1991], [Leggett et Williams, 1988] et [Coltell *et al.*, 1999]. Nos améliorations augmentent les performances du classificateur originel (on passe de 4,6% à 3,6%). Néanmoins, les intervalles de confiances (calculés avec un risque de 5%) sont importants et peuvent nuancer l'apport de nos adaptations. Nous trouvons un risque de première espèce de 23% qui peut paraitre important par rapport à ceux trouver pour d'autres applications mais qui, au vu des spécificités de la dynamique de frappe, nous semble suffisant pour démontrer une amélioration des performances dues à nos propositions.

Il est également intéressant de regarder les taux d'erreur maximum car ceux-ci donnent une bonne indication sur la possibilité d'adapter ce système à un cas réel. En effet, même avec un très bon taux d'erreur moyen, si un utilisateur sur dix ne réussit jamais à entrer, le système sera inutilisable. Or cette méthode a des taux maximums très élevés qui indique que la dégradation des performances provient de quelques utilisateurs pour lesquels on obtient des taux d'erreur très importants. Le problème est donc de réduire considérablement les taux d'erreur maximum puisque c'est sur eux que ce concentre la majorité des erreurs.

Pour expliquer ces taux d'erreur importants, il est possible de s'intéresser au comportement d'un utilisateur problématique. Ce comportement est présenté sur la Figure 24 et la Figure 25. Nous présentons, sur ces figures, le comportement d'un utilisateur et d'un imposteur avec la première version du classificateur afin d'améliorer la lisibilité des figures. La Figure 24 présente deux essais d'un utilisateur pour entrer dans le système avec son identifiant correct. On observe que les temps

extraits, de ses deux tentatives, se trouvent bien dans, le tunnel, celui-ci n'aura donc pas de difficulté pour accéder au système. La Figure 25 représente deux attaques d'imposteurs contre l'utilisateur. L'imposteur 1 échoue car la majorité des temps sont à l'extérieur du tunnel, par contre l'imposteur 2 est accepté à tord car ses temps se trouvent en majorité dans le tunnel. Pourtant une comparaison de son tracé avec le tunnel traduit le statut d'imposteur.

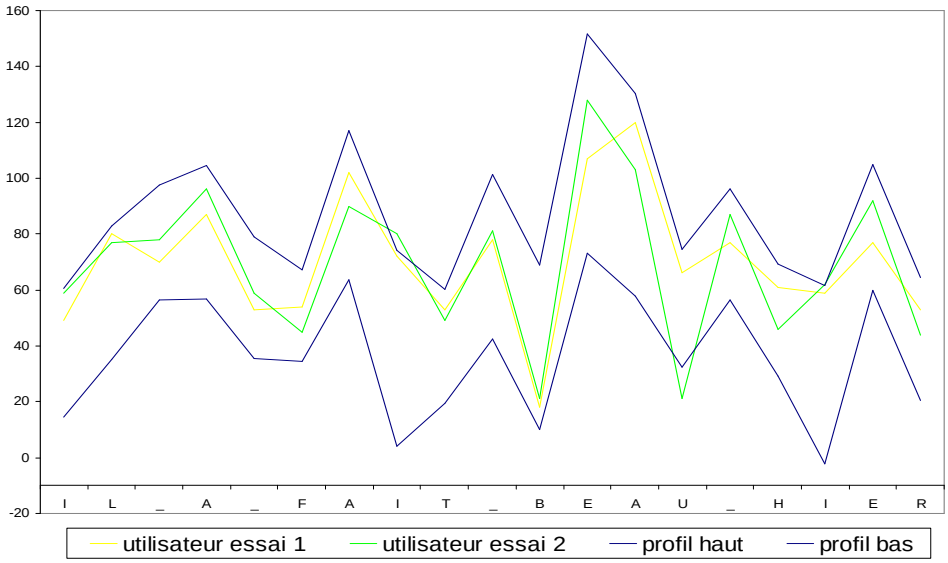


Figure 24 : Temps P-P d'un utilisateur contre lui-même

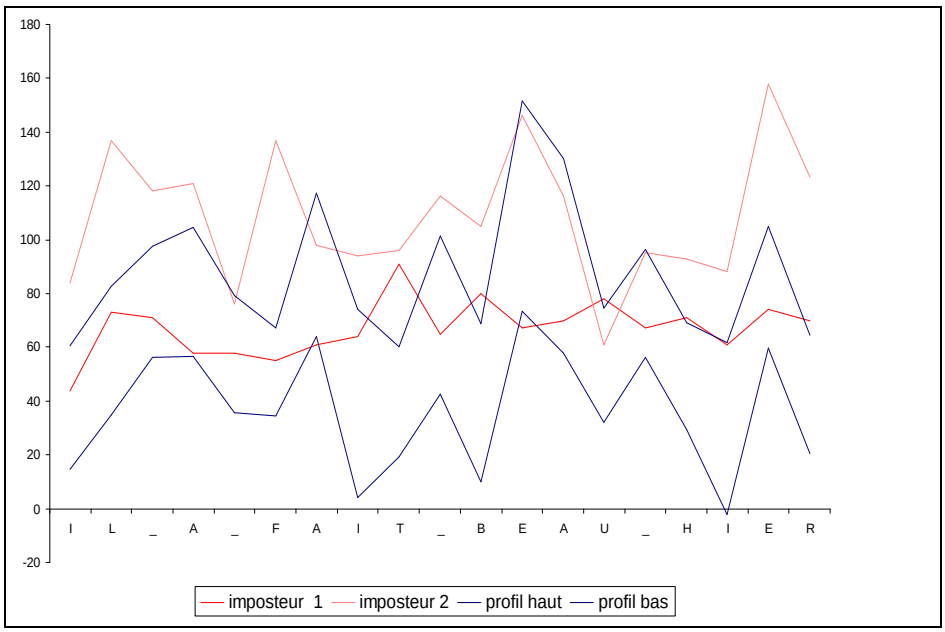


Figure 25 : Temps P-P de deux imposteurs contre un utilisateur

Il semble donc préférable de coupler ce classificateur avec d'autres méthodes utilisant d'autres caractéristiques complémentaires pour obtenir de meilleures performances.

3.2.2. Rythme de frappe

Comme pour la musique où le rythme de la mélodie est défini par la durée des notes (représentée par des symboles rondes, blanches, noires…), il est possible de s'intéresser au rythme de la frappe de l'utilisateur, c'est-à-dire non plus aux valeurs numériques des temps pris séparément, mais à la durée des temps les uns par rapport aux autres. Pour ce faire, il existe plusieurs méthodes envisageables. Celle que nous proposons consiste à classer les temps dans 5 classes distinctes : très court, court, moyen, long et très long. Pour classer les temps dans chaque catégorie, nous avons essayé plusieurs moyens :

1. Le premier a été de fixer des seuils afin de délimiter les différentes classes de temps donnés en millisecondes.
    1. t>200 classe 1
    2. 200>t>100 classe 2
    3. 100>t>70 classe 3
    4. 70>t>30 classe 4
    5. 30>t classe 5

L'inconvénient de cette méthode réside dans le choix des valeurs limites des intervalles de discrétisation indépendamment de la vitesse moyenne de frappe de l'utilisateur.

2. Le second moyen testé a été de classer les temps au sein d'une même observation, en fonction de sa durée par rapport aux autres. Là encore, le temps est classé en fonction de sa durée, mais cette fois, les classes ne sont pas délimitées par des seuils fixes, mais uniquement en fonction de la répartition des temps.
    - 1/10 des plus courts classe 1
    - 1/3 des plus courts classe 2
    - 2/3 des plus courts classe 3
    - 3/4 des plus courts classe 4
    - 1/4 des plus longs classe 5

Ensuite, quelle que soit la méthode choisie, le vecteur de caractéristiques contient les classes et les temps. Le profil d'un utilisateur est un vecteur de caractéristiques unique obtenu en retenant la classe la plus souvent affectée à un temps lors des observations issues de l'apprentissage. Pour des raisons de cohérence, ce profil n'est calculé que sur un type de temps, nous avons choisi de travailler exclusivement avec les temps RP car ceux-ci se sont révélés les plus discriminants. Pour comparer une observation avec un profil, il est nécessaire de construire une distance en fonction de la classe où a été affecté un temps et de sa classe dans le profil. Nous avons choisi de définir cette distance comme la différence entre les indices des classes. Par exemple, si un temps est classé dans la classe 4 alors qu'il était dans la 2 dans le profil, la distance vaudra 2. On réalise ensuite la somme des distances pour obtenir le score final qui sera ensuite comparé avec un seuil afin de décider si l'observation doit être acceptée ou non.

Les performances du classificateur présentées dans le Tableau 4 montrent des performances globales proches de celle du classificateur précédent basé sur des mesures statistiques. Les deux variantes présentent des performances quasiment identiques.

Tableau 4 : Performance du classificateur basé sur le rythme de frappe

Méthode	TEE'	TFA moyen	TFR moyen	TFA max	TFR max	Intervalles de confiance à 95% Pour le TEE'
Rythme de frappe (méthode 1)	3,4%	3,5%	3,4%	11,5%	18,0%	[2,7 ; 6,6]
Rythme de frappe proportion (méthode 2)	3,9%	3,6%	4,0%	10,9%	18,0%	[2,7 ; 6,6]

3.2.3. Mesure du désordre

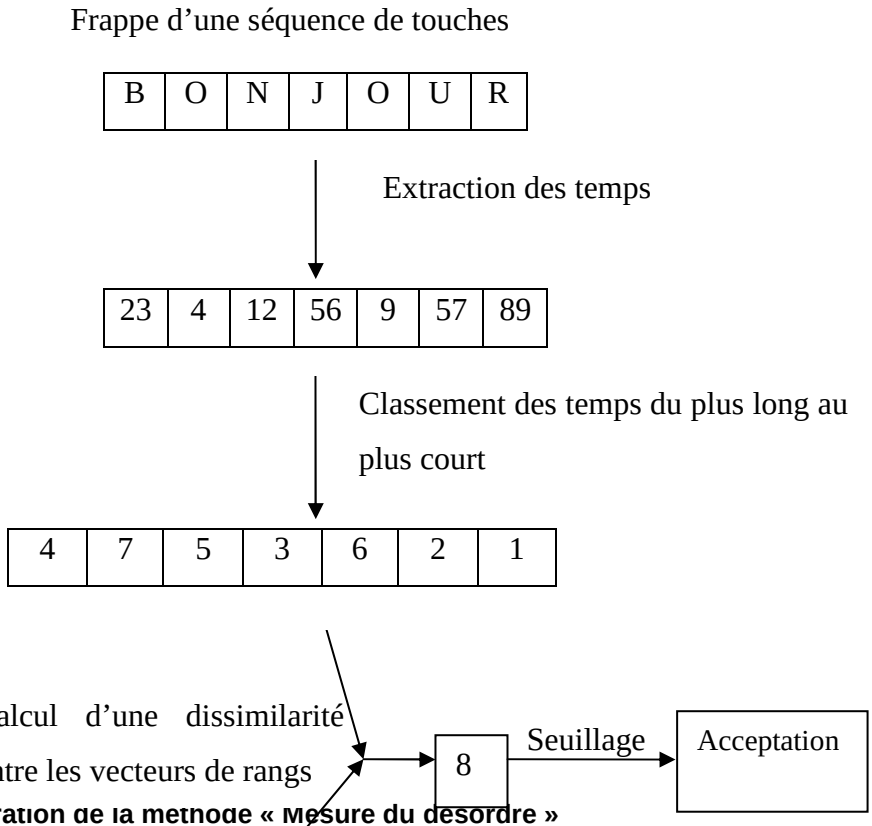


Figure 26 : Illustration de la méthode « Mesure du désordre »

Une autre façon de comparer deux dynamiques de frappe au clavier est de s'intéresser à l entifier quel temps est plus court que désordre entre deux vecteurs. Cette méthoc *l.*, 2002]. Pour mesurer cette différence dans l'ordre des temps, les temps au sein de chaque observation sont ordonnés du plus long au plus court (Figure 26).

Le profil correspond à nouveau à un seul vecteur contenant la moyenne des rangs pour chaque temps calculé sur les séquences issues de l'apprentissage. Pour déterminer le score d'une observation, il existe plusieurs méthodes : la première (et la plus simple) consiste à calculer une distance euclidienne entre le profil et une observation. La seconde méthode consiste à calculer le coefficient de corrélation de Spearman (Equation (20)) entre deux classements. Ce coefficient est conseillé lorsqu'il s'agit de comparer des rangs. Avec r_i^1 le rang du temps i dans le classement *1* et n le nombre de temps dans un classement.

$$r_{Sp} = 1 - \frac{6 * \sum_{i=1}^{n} (r_i^1 - r_i^2)^2}{n * (n^2 - 1)} \qquad \textbf{(20)}$$

Ces deux mesures de similarité ne nous ont pas paru entièrement satisfaisantes, car dans la dynamique de frappe, il arrive fréquemment qu'un temps soit beaucoup plus long dans une observation que dans le profil à cause d'une légère hésitation de l'utilisateur. Avec les mesures précédentes, un tel phénomène fausse non seulement un rang, mais perturbe également tous les rangs suivants. Nous avons implémenté une méthode corrigeant ce problème

Notre méthode (Figure 27) consiste à classer les temps par rangs décroissants dans l'observation. On étudie ensuite les temps en partant du plus long vers le plus court dans l'observation. Le score est initialisé à 0. Pour un temps dont le rang est *ro* dans l'observation et *rp* dans le profil, on ajoute au score |*rp*-*ro*| et on va corriger les rangs des temps compris entre *ro* et *rp* en les décalant de 1. On continue jusqu'à avoir passé en revue tous les temps.

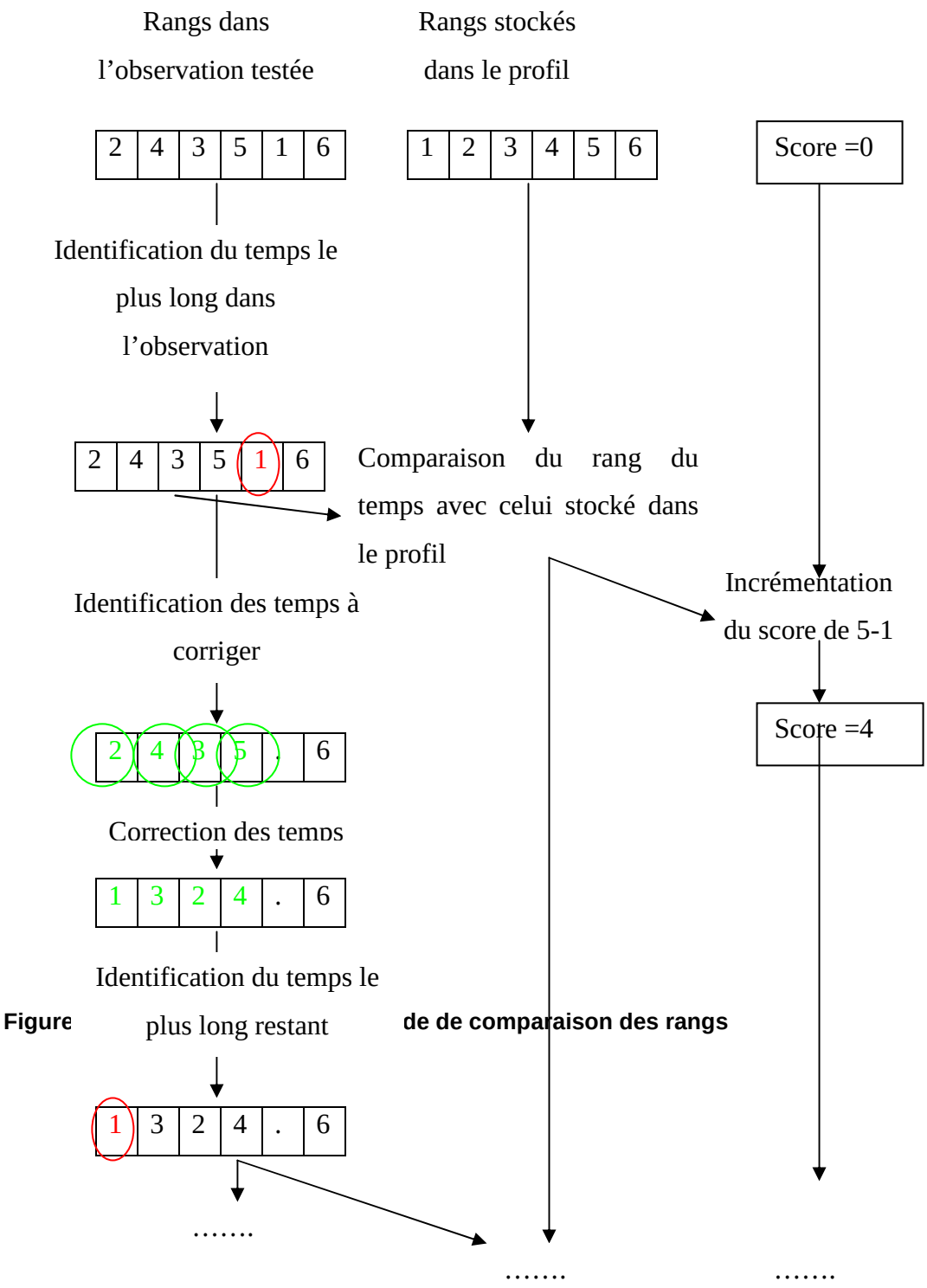


Figure de de comparaison des rangs

Tableau 5 : Performances du classificateur basé sur les rangs des temps

Méthode	TEE'	TFA moyen	TFR moyen	TFA max	TFR max	Intervalles de confiance à 95% Pour le TEE'
Rang Spearman	4,7%	4,7%	4,7%	19,0%	24,0%	[1.3 ; 8]%
Rang "corrigé"	3,6%	3,6%	3,6%	11,5%	18,0%	[1,2 ; 6]%

Les performances du classificateur sont présentées sur le Tableau 5. Ce dernier confirme encore que la plupart des classificateurs offrent des performances comparables les uns aux autres (4%). Ce tableau témoigne également du fait que l'amélioration que nous proposons pour calculer la dissimilarité entre vecteur permet un net gain de performances (mais toujours significatif avec un risque de première espèce de 30%)

3.2.4. Fusion des classificateurs

Dans le chapitre précédent, nous préconisons d'introduire une phase de fusion dans le système d'authentification biométrique. Cette section décrit les expérimentations menées pour valider cette affirmation.

3.2.4.1. Comparaison des différents classificateurs

Le Tableau 6 et la Figure 28, présente une comparaison des performances des classificateurs par utilisateur sur une base de 13 individus.

Tableau 6 : Comparaison des classificateurs

	TEE rang	TEE statistique	TEE rythme
Utilisateur 1	0,13%	3,37%	0,56%
Utilisateur 2	1,38%	3,69%	1,38%
Utilisateur 3	14,76%	4,07%	6,32%
Utilisateur 4	0,54%	0,06%	0,54%
Utilisateur 5	6,99%	0,42%	4,37%
Utilisateur 6	4,16%	10,80%	6,72%
Utilisateur 7	9,76%	6,20%	14,45%
Utilisateur 8	0,52%	0,63%	0,59%
Utilisateur 9	1,20%	1,57%	2,64%
Utilisateur 10	0,44%	1,00%	1,07%
Utilisateur 11	1,49%	5,84%	0,24%
Utilisateur 12	1,77%	4,50%	2,54%
Utilisateur 13	3,54%	4,90%	3,20%
Moyenne	3,59%	3,62%	3,43%
Maximum	14,76%	15,80%	14,45%
Minimum	0,13%	0,00%	0,00%

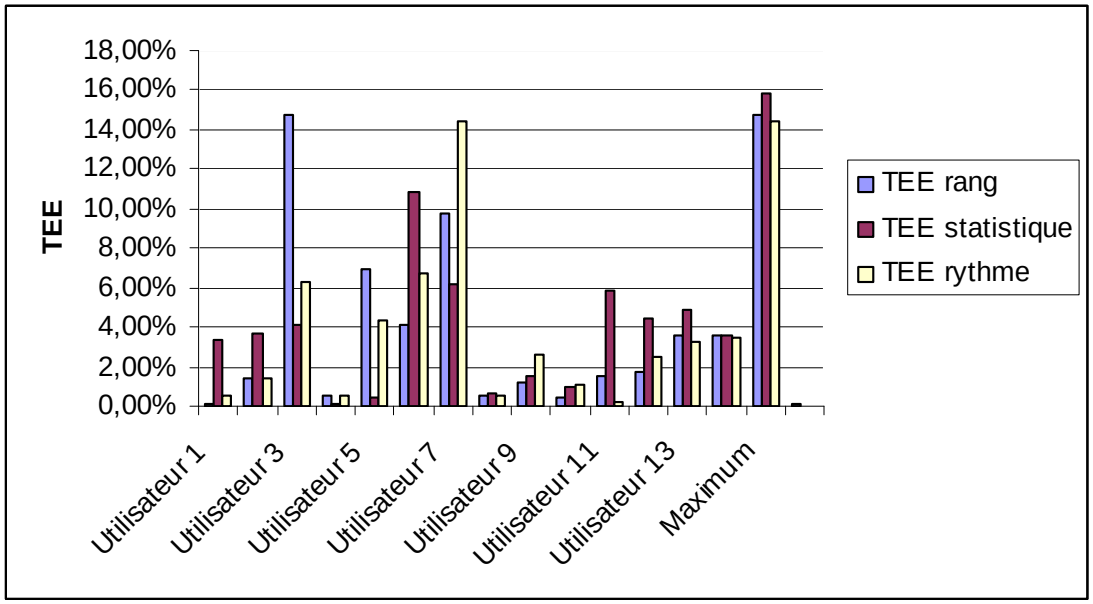


Figure 28 : Comparaison des trois classificateurs

On observe les mêmes phénomènes sur toutes les méthodes : une moitié des utilisateurs ont de bons taux d'erreur, c'est-à-dire un TEE' inférieur à 2,5% ; une très grande majorité des utilisateurs présentent un TEE' inférieur à 5% ce qui reste acceptable ; enfin, un ou deux utilisateurs obtiennent des taux d'erreur extrêmement

élevés c'est-à-dire avec un TEE proche (voire dépassant) les 10%. Pour la méthode statistique, on observe quatre utilisateurs ayant des taux d'erreur supérieurs à 5% pour le TFR et également quatre pour le TFA. Un utilisateur obtient même un TFA très élevé de 22%. Après vérification, cet utilisateur a pour particularité d'avoir une vitesse de frappe élevée en moyenne mais avec une grande variation provoquant l'apparition d'une grande tolérance pour les imposteurs.

La méthode des rangs obtient des performances globales très proches de la méthode statistique. Par contre, les répartitions des taux d'erreur diffèrent significativement. Seuls quatre utilisateurs obtiennent des taux d'erreur supérieurs à 2,5%, avec parmi eux, l'utilisateur 5 qui obtient un très bon taux d'erreur avec la méthode statistique.

La méthode se basant sur l'étude du changement de rythme produit le même phénomène que les autres méthodes : les taux d'erreur importants sont concentrés sur quelques utilisateurs. Les résultats de ces tests montrent que les différentes méthodes donnent des résultats assez proches en terme de performances globales, mais que les erreurs ne proviennent pas des mêmes utilisateurs. La fusion devrait donc apporter un gain significatif de performances.

3.2.4.2. Performance avec la fusion

Suite aux études présentées dans les chapitres 1et 2, nous avons choisi de mettre en place une fusion « *Somme* » après une normalisation des scores fournis par les différents classificateurs. Il s'agit maintenant de vérifier si ce choix est judicieux dans le cadre de la dynamique de frappe.

Le premier test que nous avons réalisé avait pour objectif de comparer entre eux les différents moyens de normaliser les scores. L'opérateur Somme est utilisé et les résultats de nos expérimentations sont présentés dans le Tableau 7.

Tableau 7 : Comparaison des différentes méthodes de normalisation (appliquées à l'opérateur Somme)

Normalisation	TFA	TFR	TEE'
(Score-min)/(max-min)	1,8%	2,5%	2,1%
Score/max	2,0%	2,0%	2,0%
z-score	1,8%	1,7%	1,8%

Ce tableau montre que la normalisation a une influence sur les performances du système. Cela est dû aux différences d'aplatissement des intervalles. On observe que les meilleurs résultats sont obtenus à l'aide d'une normalisation à l'aide du calcul du z-score. C'est donc cette méthode de normalisation qui sera utilisée pour cette application.

Tableau 8 : Résultats des différents opérateurs de fusion

Méthode	TFA	TFR	TEE'	Intervalles de confiance à 95% Pour le TEE'
Vote unanime	1,17%	7,92%	4,55%	[2,5 ; 6,5] %
Vote majoritaire	2,39%	2,15%	2,27%	[2,1 ; 4,4]%
Vote unique	7,60%	0,54%	4,07%	[1,5 ; 6,5]%
Produit	2,00%	2,00%	2,00%	[1 ; 3]%
Maximum	3,62%	3,61%	3,62%	[1,6 ; 5,6]%
Minimum	3,62%	3,62%	3,62%	[1,6 ; 5,6]%
Médian	3,34%	3,39%	3,37%	[1,5 ; 5,7]%
Somme	1,81%	1,69%	1,75%	[0,7 ; 2,8]%

Une fois la méthode de normalisation des scores de classificateurs choisie, il est nécessaire de sélectionner l'opérateur de fusion à utiliser. Les résultats des différents opérateurs de fusion sont présentés dans le Tableau 8. La première conclusion ressortant de cette étude est que la fusion améliore considérablement les résultats par rapport à chaque classificateur pris séparément. Nous confirmons ainsi les résultats énoncés dans les chapitres précédents ainsi que ceux classiquement énoncés dans la littérature. Ces tests montrent que même si les classificateurs travaillent à partir des mêmes données de base (données brutes), les caractéristiques constituées ensuite sont suffisamment différentes pour justifier une fusion. Les opérateurs les plus performants sont *le vote majoritaire*, le *produit* et la *somme*. Au niveau performance, l'opérateur *somme* est le plus performant avec un TEE' de 1,75%, ensuite vient le *produit* avec un TEE' de 2,00% et *le vote majoritai*re avec un TEE' de 2,27%. On ne peut pas uniquement comparer ces méthodes selon leur performance, il est également intéressant d'analyser les connaissances nécessaires sur les données et sur les seuils utilisés par chaque méthode. La méthode du *vote*

majoritaire nécessite la détermination de 3 seuils, un par méthode utilisée. La méthode *somme* nécessite de connaître une estimation de la moyenne et de la variance des scores pour la normalisation ainsi que la sélection d'un seuil pour la décision. Le *produit* peut être utilisé en utilisant un unique seuil pour la décision. La méthode *produit* même si elle est moins performante que la méthode *somme*, est donc applicable plus facilement surtout dans certains cas comme par exemple pour les mots de passe de longueur très variable. Il est, en effet, dans ce cas difficile de normaliser les scores avant la fusion. Pour notre application, nous avons décidé d'utiliser l'opérateur *somme* pour sa simplicité et ses performances globalement meilleures.

Les résultats détaillés obtenus avec l'opérateur *somme* finalement retenu sont présentés dans le Tableau 9 et la Figure 29 . Pour montrer l'importance de la fusion, nous nous sommes limités dans un premier temps à fixer un poids égal aux trois classificateurs. La manière dont nous adapterons ensuite ces poids est présentée dans la section suivante. Ce tableau qui est à comparer avec le Tableau 6 montre une baisse globale des taux d'erreur par utilisateur par rapport à chaque méthode prise séparément (significatif avec un risque de première espèce à 10%) Mais pour certains utilisateurs, la fusion obtient de moins bons résultats que la méthode la plus performante (utilisateurs 5 et 7). On peut remarquer une baisse significative des taux d'erreur maximums qui descendent à 8%. La fusion est donc une composante essentielle de l'architecture du système biométrique.

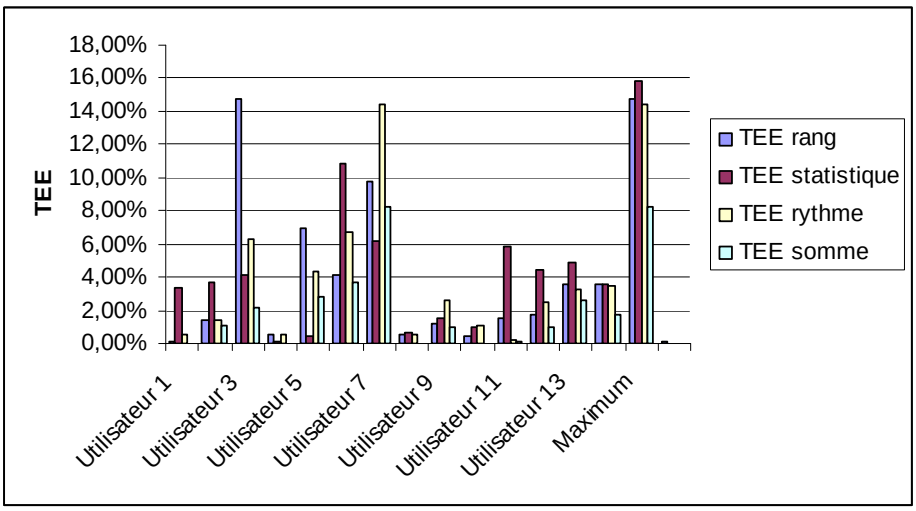


Figure 29 : Performance de l'opérateur *somme*

Tableau 9 : Résultats détaillés pour l'opérateur *Somme* sur 13 individus

	TFA	TFR	TEE'
Utilisateur 1	0,00%	0,00%	0,00%
Utilisateur 2	0,25%	2,00%	1,13%
Utilisateur 3	4,38%	0,00%	2,19%
Utilisateur 4	0,00%	0,00%	0,00%
Utilisateur 5	0,60%	5,00%	2,80%
Utilisateur 6	7,48%	0,00%	3,74%
Utilisateur 7	8,39%	8,00%	8,20%
Utilisateur 8	0,00%	0,00%	0,00%
Utilisateur 9	0,00%	2,00%	1,00%
Utilisateur 10	0,00%	0,00%	0,00%
Utilisateur 11	0,24%	0,00%	0,12%
Utilisateur 12	0,93%	1,00%	0,97%
Utilisateur 13	1,20%	4,00%	2,60%
Moyenne	1,81%	1,69%	1,75%
Maximum	8,39%	8,00%	8,20%
Minimum	0,00%	0,00%	0,00%

3.2.5. Bilan sur le choix des classificateurs

Les trois classificateurs que nous avons utilisés ont été choisis pour leur simplicité. Ces derniers ne nécessitent aucun paramètre pour fonctionner, seul le

seuil de décision est nécessaire. Notre architecture propose une fusion des classificateurs avec des poids normalisés pour que leur somme soit égale à 1. Au final, il faut déterminer un seuil pour la décision finale et 2 poids (le troisième étant déduit) nécessaires pour la fusion. Pour paramétrer le système complètement, les valeurs de ces paramètres et seuils doivent être déterminés. Nous avons choisi de régler ces paramètres individuellement pour chaque utilisateur. Les méthodes que nous proposons pour cela sont présentées dans la partie suivante.

3.3. Personnalisation du système

3.3.1. La mise à jour du profil

Lors de l'utilisation de méthodes de biométrie comportementale, le comportement d'un utilisateur évolue en fonction de son habitude d'utilisation du dispositif d'acquisition. Pour la dynamique de frappe, ce comportement évolue, non seulement en fonction de l'évolution des compétences de l'utilisateur mais aussi en fonction de l'apprentissage qu'il fait de la séquence qu'il doit entrer. Une mise à jour du profil est donc indispensable.

3.3.1.1. Méthode de mise à jour du profil

La mise à jour du profil doit être réalisée après une authentification réussie. La méthode de mise à jour choisie est le re-calcul du profil sur les dix dernières séquences acceptées. Avant de réaliser la mise à jour, nous avons vu au chapitre 2 que nous préconisons de vérifier que la séquence acceptée ne comporte pas d'anomalies (comme par exemple des pauses ou des hésitations pour la dynamique de frappe) à l'aide d'heuristiques simples.

Notre base d'utilisateurs comprend un nombre important d'utilisateurs ayant fourni des séquences durant un grand laps de temps (20 utilisateurs sur 40 ont fournis plus de 50 séquences sur une durée de 6 mois minimum). Nous avons donc pu examiner en détail, l'influence de la mise à jour du profil sur les performances. Les résultats sont calculés sur la totalité de la base (40 utilisateurs).

Le premier test que nous avons conduit est de suivre en détail l'évolution des scores sur les deux utilisateurs ayant fournit le plus de séquences.

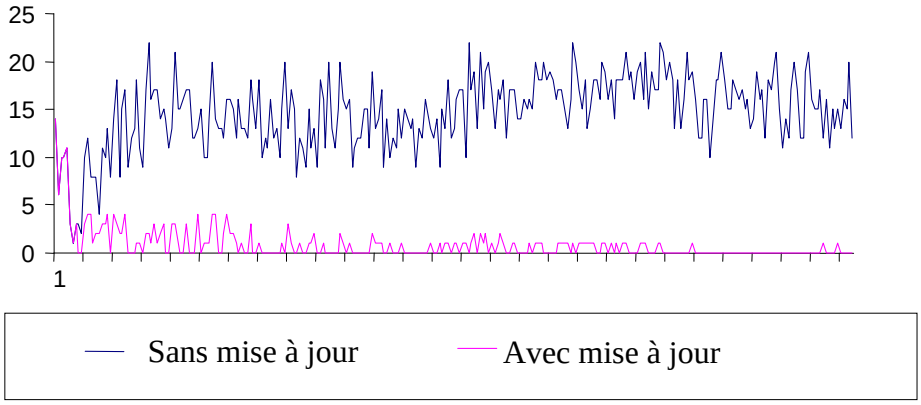


Figure 30 : Evolution des scores pour l'utilisateur 13

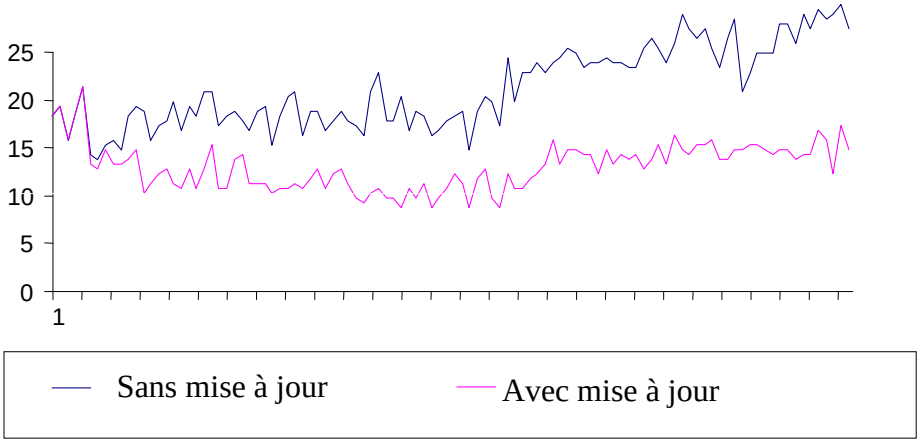


Figure 31 : Evolution des scores pour l'utilisateur 21

L'évolution des scores pour les deux utilisateurs que nous avons choisis est présentée sur la Figure 30 et la Figure 31. Nous voyons pour l'utilisateur 13 une stabilisation rapide du profil et une très grande différence entre le calcul des scores avec mise à jour et sans mise à jour. Pour l'utilisateur 21, la stabilisation n'intervient pas mais la mise à jour apporte quand même un net gain de performance.

Nous allons maintenant comparer la performance des différentes stratégies de mise à jour sur notre population. Les stratégies considérées sont les suivantes :

- Pas de mise à jour.
- Toutes les séquences valides sont insérées dans le profil et utilisées par la suite.
- Seules les *n* dernières séquences valides sont introduites dans le profil (*n* a été fixé à 10).

- Toutes les séquences valides sont introduites dans le profil avec un poids décroissant suivant la chronologie des séquences (1, pour la dernière, et *(1-(n-1/n))* pour la n^{eme} séquences avant la dernière.

Tableau 10 : Performances selon la stratégie de la mise à jour du profil

	TEE'	TFR	TFA	TFR Maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Sans mise à jour	13,5%	15,0%	12,1%	58,3%	67,5%	[9,5 ; 17,5]%
Tous les logins	10,5%	13,0%	8,1%	58,3%	67,5%	[7,5 ; 16]%
Les dix dernières séquences	6,0%	8,0%	4,1%	58,3%	38,5%	[3,0 ; 9,0]%
Avec mise à jour et avec un poids	6,4%	9,0%	3,8 %	58,3%	38,5%	[3,0 ; 9,8]%

Les performances observées sont présentées dans le Tableau 10. Il est ainsi confirmé que la mise à jour apporte un gain de performance important (significatif avec un risque de première espèce de 10%) et sensible donc essentielle dans tous les systèmes biométriques basés sur des caractéristiques biométriques qui évoluent au cours du temps. Dans nos expérimentations, la méthode de mise à jour qui donne le meilleur résultat consiste à prendre uniquement les dix dernières séquences. C'est cette méthode que nous retenons pour notre application.

3.3.2. Influence du matériel

La dynamique de frappe est une méthode biométrique qui peut être dépendante du matériel (le clavier).

Il est intéressant d'examiner ce qui peut se passer si un utilisateur doit changer de matériel. Pour ce faire, nous avons demandé à 6 utilisateurs de taper une séquence de touches 30 fois sur un clavier standard et 20 fois sur un clavier de

portable. La création du profil est réalisée sur les dix premières séquences entrées sur le clavier standard.

Tableau 11 : Influence du clavier et de la mise à jour

	Nombre de séquences rejetées			
	Sans mise à jour		Avec mise à jour	
	clavier standard	portable	clavier standard	portable
Utilisateur 1	0	7	0	3
Utilisateur 2	4	5	3	1
Utilisateur 3	0	0	0	1
Utilisateur 4	0	4	0	1
Utilisateur 5	3	2	2	0
Utilisateur 6	2	7	2	4
total	9	25	7	10
TFR	7,5%	20,8%	5,8%	8,3%

Nous pouvons voir sur le Tableau 11 que le changement de clavier perturbe considérablement l'authentification des utilisateurs, le TFR passe de 7,5% à 20,8%. Par contre, avec l'implémentation de la mise à jour du profil, il est possible de conserver un fonctionnement correct même si ce dernier est un peu perturbé temporairement.

Cette étude, même si elle porte sur peu d'utilisateurs, nous permet de valider l'intérêt d'une mise à jour progressive du profil lors du changement de matériel.

Il est à noter que nous avons imposé aux utilisateurs de ne pas choisir de séquences comportant des touches dont l'emplacement change suivant le clavier (pavé numérique par exemple) car, dans ce cas, on ne peut pas échapper à un nouvel enregistrement.

3.3.3. Personnalisation des paramètres du système de décision

Les classificateurs que nous avons choisis ne nécessitent aucun paramètre, par contre la fusion nous permet de spécifier des poids attribués à chaque classificateur (w_i). Dans notre application de dynamique de frappe, il est même possible d'associer un poids à chaque type de temps extraits lors de la frappe (PP, PR...).

Le score final est donc calculé à l'aide des équations (21) et (22) .

$$ScoreFinal(FSC) = \sum_i w_i * score_i \qquad \textbf{(21)}$$

$$Score_i = w_{i,pp} * Score_{i,pp} + w_{i,pr} * Score_{i,pr} + w_{i,rp} * Score_{i,rp} + w_{i,rr} * Score_{i,rr} \quad \textbf{(22)}$$

Avec $score_{i,xy}$ le score calculé à l'aide du classificateur *i* sur les temps *XY,* et $w_{i,xy}$ le poids associé. Chaque jeu de poids est normalisé afin que la somme des poids du jeu soit égale à 1. Nous avons donc 2 poids à fixer pour les classificateurs et 9 poids (3*3) à déterminer concernant les types de temps. En plus de ces poids, il est nécessaire de fixer le seuil de décision finale.

3.3.3.1. Etude de l'influence des paramètres sur les performances

Le nombre de paramètres dont nous avons besoin dans notre méthode, même s'il est raisonnable en comparaison à d'autres méthodes, reste important. L'objectif est donc de mettre en place une procédure conforme à ce que nous préconisons dans le chapitre 2 et permettant de déterminer automatiquement l'ensemble des paramètres. Nous étudierons également la possibilité d'adapter automatiquement ces paramètres aux comportements individuels de chaque utilisateur. Les paramètres à déterminer sont les suivants :

- Seuil de décision

Pour déterminer s'il est utile de choisir un seuil de décision différent pour chaque individu, nous avons comparé deux possibilités : un seuil global fixé pour tous et un seuil individualisé pour chaque utilisateur choisi de manière à minimiser le TFR+TFA calculé en utilisant le contenu du profil de l'utilisateur et la base de référence. Les résultats sont présentés dans le Tableau 12.

Le gain de performance entre seuil global et seuils individuels est énorme avec un gain de 1/3 (mais toujours significatif avec un risque de 20%).

Tableau 12 : Seuils individuels contre seuil global

	TEE'	TFR moyen	TFA moyen	TFR maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Seuil global	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Seuils individuels	4,2%	4,1%	4,3%	20,2%	40,9%	[2,3 ; 6]%

Ce test confirme donc l'importance de mettre en place une telle fonctionnalité comme nous le recommandons dans le chapitre 2.

- Poids affectés aux classificateurs

De la même manière que précédemment, pour examiner l'intérêt d'adapter des poids de fusion à chaque utilisateur, trois tests différents ont été réalisés. Les poids sont calculés à l'aide d'une recherche exhaustive de manière à minimiser le TFR+TFA calculé en utilisant le contenu du profil de l'utilisateur et la base de référence.

1. Utilisation de poids identiques pour tous : les trois poids de fusion sont fixés à 0,33< ;
2. Utilisation de poids globaux : on utilise le jeu de poids qui fonctionne le mieux pour l'ensemble des utilisateurs (minimisation du TFA+TFR global).
3. Utilisation de poids individuels : pour chaque utilisateur, on utilise les poids qui minimisent la somme TFA+TFR.

Le Tableau 13 montre un gain de performance entre poids globaux et poids individuels moins net que dans le cas du seuil de décision (significatif avec un risque de première espèce de 25%) mais il reste suffisamment important pour justifier l'implémentation de ce module dans les systèmes biométriques (notamment au regard des taux maximums).

Tableau 13 : Poids optimaux contre poids globaux pour les classificateurs

	TEE'	TFR moyen	TFA moyen	TFA maximum	TFR maximum	Intervalles de confiance à 95% Pour le TEE'
Poids égaux	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Poids globaux	5,8%	5,8%	5,8%	58,3%	38,5%	[2,8 ; 8,8]%
Poids optimaux	4,7%	5,0%	4,5%	58,3%	38,5%	[1,7 ; 8]%

- Poids affectés à chaque temps

Il est possible de suivre la même démarche que précédemment mais pour associer des poids à chaque type de caractéristiques biométriques utilisées (les temps PP, PR, RP, RR extraits dans le cadre de la dynamique de frappe). Les performances en utilisant un tel procédé sont fournies dans le Tableau 14. Nous avons défini un jeu de poids pour chaque type de temps et chaque classificateur. Pour chaque classificateur la somme des poids est normalisée à 1. Nous avons donc au total neuf poids à calculer.

Tableau 14 : Influence de l'affectation de poids différents à chaque type de temps contenus dans les vecteurs de caractéristiques

	TEE'	TFR moyen	TFA moyen	TFA maximum	TFR maximum	Intervalles de confiance à 95% Pour le TEE'
Poids égaux pour chaque temps	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Poids différenciés et individuels	5,2%	5,3%	5,1%	58,3%	38,5%	[2,0 ; 8,4]%
Poids différenciés globaux	5,4%	5,4%	5,5%	58,3%	38,5%	[2,4 ; 8,4]%

Nous observons une nette amélioration lors de l'utilisation de poids différenciés (significatif avec un risque de première espèce de 25%) pour chaque type de temps par rapport à l'affectation de poids similaires pour tous les types de temps. Mais la différence de performances entre une détermination des poids globale et individualisée est très faible (non significatif). Nous ne jugeons donc pas nécessaire d'essayer de déterminer ces poids de façon individuelle mais utiliserons des poids simplement différenciés en fonction de chaque type de temps (PP, PR, RP, RR).

Tableau 15 : Poids individualisés retenus pour chaque temps

	Temps PP	Temps PR	Temps RR	Temps RP
Méthode des rangs	0,3	0,1	0,3	0,3
Méthode statistique	0,2	0,4	0,2	0,2
Méthode du rythme	0,2	0,4	0,2	0,2

Les poids que nous avons retenus pour les différents types de temps sont présentés sur le Tableau 15. Les types de temps peuvent être divisés en deux groupes dans le premier, on trouve les trois temps prenant en compte le temps entre deux touches (PP, RP, RR) et dans l'autre le temps PR qui prend en compte uniquement le temps de pression des touches. Les temps du premier groupe se voient attribués un poids identique montrant qu'ils ont la même importance. Les temps PR sont à part, leurs poids sont importants sauf en ce qui concerne la méthode des rangs où leur importance est très faible.

3.3.3.2. Construction du vecteur Ref

Les informations dont nous disposons grâce à la base de référence et aux profils utilisateurs, pour la détermination des paramètres de chaque utilisateur sont :

- une caractéristique représentant la séquence dans sa globalité : longueur de la séquence
- des caractéristiques représentant la vitesse de frappe de l'utilisateur
  - durée moyenne pour entrer la séquence

- moyenne des temps PP, RP, PR et RR
- maximums des temps PP, RP, PR et RR

- des caractéristiques représentant la variabilité de l'utilisateur
  - variance de la durée calculée sur les séquences d'enregistrement
  - score moyen donné par chaque classificateur sur les séquences d'enregistrement
  - variances des scores donnés par chaque classificateur calculées sur les séquences d'enregistrement
  - maximum des scores donnés par chaque classificateur calculé sur les séquences d'enregistrement

Nous disposons donc de 20 caractéristiques pour nous aider à fixer la valeur des paramètres à déterminer pour chaque utilisateur. Ces caractéristiques sont regroupées dans le vecteur *Ref*. Ce nombre est important pour que toutes ces caractéristiques soient utilisées directement par les estimateurs, et il n'est pas sûr que toutes permettent de faire le lien avec les paramètres, il sera donc probablement nécessaire de le réduire à l'aide de différentes méthodes de sélection de caractéristiques.

3.3.3.3. Détermination directe des paramètres

Afin d'appliquer les méthodes d'estimation directe présentées dans le chapitre 2, nous devons réduire la taille du vecteur *Ref*. Pour cela, nous avons utilisé l'algorithme SFFS présenté dans le chapitre 1. Cet algorithme nécessite la détermination d'une fonction de performance qui permet d'évaluer un groupe de caractéristiques. Comme nous avons un problème d'estimation, nous choisissons comme fonction de performance la somme des écarts entre les paramètres estimés et les paramètres optimaux calculés sur la base de référence en *Leave One Out*.

Tableau 16 : Comparaison des méthodes d'estimation directe des paramètres individuels

	TEE'	TFR	TFA	TFR Maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres globaux	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Paramètres optimaux	3,5%	3,3%	3,8%	20,2%	40,9%	[1,3 ; 5,7]%
3-plus proches voisins	7,3%	7,7%	6,8%	70,0%	53,3%	[4,8 ; 9,8]%
Réseau de neurone	6,5%	8,7%	4,2%	44,4%	41,8%	[3,8 ; 8,8]%

Le Tableau 16, présente les résultats de l'estimation directe de paramètres sur la totalité de notre base. Nous présentons également, dans ce tableau, les résultats obtenus avec des paramètres communs à tous les utilisateurs, et les résultats obtenus avec le meilleur jeu de paramètres. Les méthodes d'estimation directe que nous avons testées sur la dynamique de frappe sont les 3-plus proches voisins et un réseau de neurones de type perceptron multicouches à une couche cachée (avec six neurones dans la couche cachée et avec comme algorithme d'apprentissage : *Resilient Backpropagation* – RPROP [Riedmiller et Braun, 1994]). Ces méthodes sont présentées en détail dans le chapitre 2.

Les performances obtenues dans le cadre de la dynamique de frappe sont plutôt décevantes. En effet, quelle que soit la méthode employée, l'estimation directe à partir du vecteur *Ref* se révèle incapable d'approcher ne serait-ce que de loin les performances des paramètres optimaux. Les résultats sont mêmes inférieurs à ceux des paramètres globaux. Ces mauvaises performances s'expliquent par la hausse des taux d'erreur maximum, qui indique que pour certains utilisateurs, les paramètres estimés notamment le seuil de décision est manifestement très loin de la valeur assurant le bon fonctionnement du système

3.3.3.4. Création de classes de comportement

- Création des classes de comportement

Les performances des estimations directes s'étant révélées médiocres pour notre application de dynamique de frappe, nous avons choisi de procéder par création de classes de comportement afin de constituer des groupes d'utilisateurs homogènes auxquels seront associés des jeux de paramètres. Pour cela, les premières méthodes que nous avons testées créent des classes de comportement à partir des paramètres optimaux des utilisateurs présents dans la base de référence. Les classes sont déterminées par la méthode des k-means, avec k=4 choisi empiriquement après avoir effectué plusieurs tests. Les classificateurs que nous avons expérimentés (pour déterminer la classe d'un nouvel individu) sont les 3-plus proches voisins et les SVM.

Tableau 17 : Estimation des paramètres par création de classes de comportement sur les paramètres optimaux

	TEE'	TFR	TFA	TFR Maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres globaux	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Paramètres optimaux	3,5%	3,3%	3,8%	20,24%	40,92%	[1,3 ; 5,7]%
Plus proches voisins+mise en classe	5,7%	6,0%	5,5%	30,9%	57,0%	[3,3 ; 8,1]%
SVM + mise en classe	4,9%	5,1%	4,7%	28,3%	48,6%	[2,6 ; 7,2]%

Les résultats obtenus (présenté par le Tableau 17), sont bien meilleurs que ceux obtenues par estimation directe (significatif avec un risque de première espèce de 30% pour les SVM). La création de classes de comportement se révèle donc judicieuse et permet de faire bien mieux que les paramètres globaux. Les résultats obtenus demeurent néanmoins assez éloignés des performances maximales atteignables produites avec des paramètres manuels optimaux.

- Création de classe de comportement à partir des caractéristiques des utilisateurs

Analyse des données

Un algorithme de classification non supervisé travaillant sur toutes les caractéristiques de nos vecteurs *Ref* risque d'avoir des difficultés à produire des classes pertinentes compte tenu de la dimension de l'espace (dimension 20). Nous avons donc décidé de réduire la dimension de l'espace. L'objectif est de trouver les caractéristiques qui nous permettrons de réaliser les meilleures classes possibles, l'analyse en composante principale (ACP) est une solution pour atteindre notre objectif. Les valeurs des dix premières valeurs propres obtenues après l'ACP sont présentées Tableau 18.

Tableau 18 : valeurs propres de l'ACP effectué sur l'espace du vecteur *Ref*

Ordre	valeur	Valeur cumulée	% de l'inertie expliquée
1	8,6	8,6	38
2	4,2	12,8	56
3	2,7	15,6	68
4	2,3	17,9	78
5	1,3	19,3	84
6	0,9	20,2	88
7	0,6	20,9	90
8	0,5	21,4	93
9	0,4	21,8	95
10	0,3	22,1	96

Les cinq premières valeurs propres expliquent plus de 80 % de l'inertie du système. Ainsi seuls les cinq premiers axes factoriels seront donc conservés pour représenter un profil d'utilisateur.

Un avantage important de l'analyse factorielle est de permettre de représenter en faible dimension, les données qui sont définies dans un espace de dimension élevée. La Figure 32 représente le premier plan factoriel. Ce plan explique seulement 56% de l'inertie initiale, mais donne une bonne première approximation de la dispersion des données. Chaque profil d'utilisateur est représenté par un point. Nous pouvons facilement voir sur cette représentation, la division de la population d'utilisateurs en trois classes apparentes. Une classe contient la majorité des utilisateurs : nous supposons que ces utilisateurs partageront des paramètres communs. Les deux autres classes regroupent des utilisateurs avec différents comportements et peuvent donc avoir besoin de paramètres plus spécifiques.

L'algorithme des k-means (avec k=3) est ensuite utilisé pour produire les classes à partir des cinq composantes principales obtenues préalablement. Cet algorithme simple donne une bonne classification sur des espaces de faible dimension.

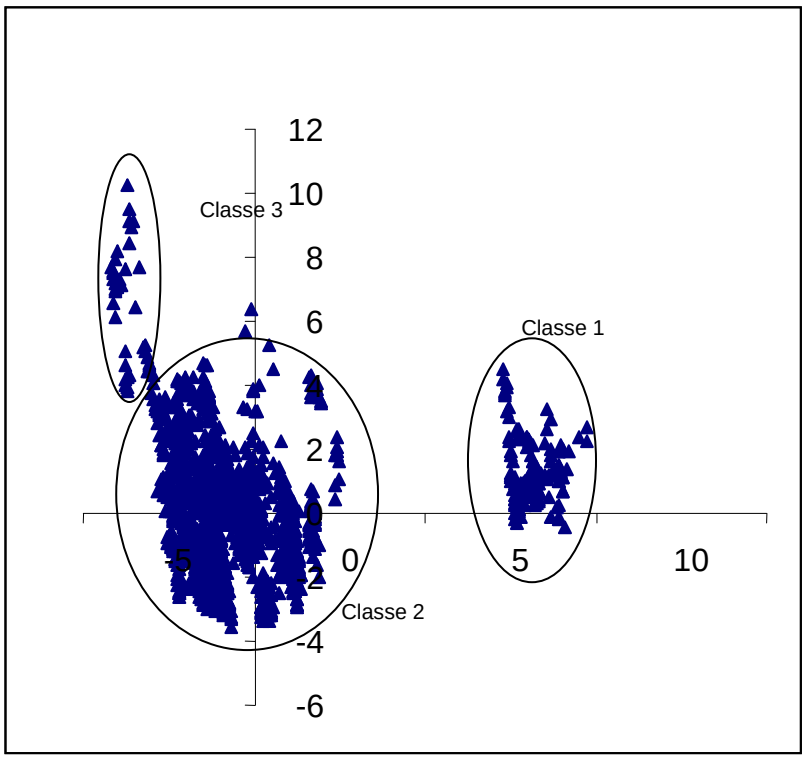


Figure 32 : Premier plan factoriel utilisé lors de la création des classes

En utilisant toujours la base de référence, le jeu de paramètres optimal est calculé pour chacune des trois classes d'individus. Les valeurs déterminées par cette méthode sont présentées Tableau 18 .

Tableau 19 : Effectifs et paramètres des classes d'utilisateurs

	Seuil de décision	Poids de la méthode statistique	Poids de la méthode du désordre	Nombre de profils dans chaque classe
classe 1	0,9	0,1	0,3	340
classe 2	1	0	0,5	1322
classe 3	1,2	0,2	0,4	156

Il est possible de faire les commentaires suivants à partir du Tableau 19 :

- La classe 2 est la plus grande classe. Cette classe est caractérisée par un seuil moyen et par une adéquation plus forte de la méthode de

discrétisation des temps et la méthode de mesure de désordre, plutôt que la méthode statistique.

- La classe 1 regroupe les utilisateurs avec un seuil de sécurité bas. Cette classe semble donc contenir des profils caractérisés par leur stabilité. C'est sur cette classe que l'on retrouve les poids les plus bas pour la méthode de mesure de désordre.
- La classe 3 semble comprendre les utilisateurs avec de grandes variations dans leur profil car le seuil de sécurité est élevé.

Tableau 20 : Performance des méthodes d'estimation des paramètres basées sur la création de classes de comportement sur les vecteurs *Ref*

	Paramètres globaux		Paramètres individualisés	
	TFA	TFR	TFA	TFR
Classe 1	0%	0,1%	0%	0%
Classe 2	1,8%	5,8%	2,8%	3,3%
Classe 3	0%	1,9%	0%	1,9%
Total	1,8%	5,3%	1,7%	2,1%

Le Tableau 20 montre une nette amélioration des performances comparées à l'utilisation des paramètres globaux. Les résultats sont améliorés pour toutes les classes. Les taux d'erreur obtenus sont très bons pour une méthode d'analyse de la dynamique de frappe. Cependant, ces taux d'erreur cachent le fait que l'erreur n'est pas calculée de la même façon que pour les autres méthodes. Ici on calcule les taux sur une classe, au lieu de calculer la moyenne des taux d'erreur obtenus individuellement pour chaque utilisateur. Elle tend à augmenter l'influence des utilisateurs ayant fourni beaucoup de séquences au détriment de ceux en ayant fournies peu. Or les utilisateurs ayant fourni le plus de séquences sont ceux qui obtiennent les meilleures performances. Dans notre base, nous avons identifié trois

utilisateurs ayant de très mauvaises performances (TEE>30%), ils ont donné seulement quelques séquences (entre 20 et 40) aussi leur influence est petite.

Afin de corriger ce biais, le Tableau 21 présente les résultats reformulés de la façon classique (moyenne des taux des utilisateurs), nous observons quand même une légère amélioration par rapport aux paramètres globaux (le TEE est amélioré de 0,5%) mais le calcul du risque de première espèce (40%) nous indique cette fois ci que l'amélioration est non significative.

Tableau 21 : Estimation des paramètres par création de classes de comportement sur les caractéristiques des utilisateurs

	TEE'	TFR	TFA	TFR Maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres globaux	6,1%	8,0%	4,1%	58,3%	38,5%	[3,1 ; 9,1]%
Paramètres optimaux	3,5%	3,3%	3,8%	20,2%	40,9%	[1,3 ; 5,7]%
Plus proches voisins+mise en classe	5,5%	6,4%	4,6%	55,0%	47,0%	[3,1 ; 7,9]%

3.3.3.5. Bilan de l'estimation des paramètres

Les méthodes préconisées pour estimer des paramètres individualisés provoquent une amélioration nette des performances par rapport à l'utilisation de paramètres globaux. La méthode donnant les meilleurs résultats est celle basée sur une création des classes de comportement suivant les paramètres optimaux observés sur la base de référence, avec ensuite une utilisation des SVM pour déterminer la classe du nouvel individu. Ces expérimentations confirment donc nos préconisations de mise en place d'un module d'adaptation des paramètres des individus.

3.3.4. Détermination des profils inconsistants

Nous avons déjà constaté que la grande majorité des erreurs étaient concentrées sur un petit nombre de personnes. Nous avons conseillé (chapitre 2) pour résoudre ce problème, de détecter les utilisateurs susceptibles de poser ce genre de problème.

Pour la dynamique de frappe, nous avons utilisé des heuristiques afin d'écarter les profils manifestement inconsistants. Ces heuristiques utilisent la présence d'anomalies comme :

- La présence de longues pauses (temps PP >2s) lors de la frappe des séquences qui composent le profil
- La présence de temps d'appui sur une touche anormalement longs (temps PR > 1s)

Tout profil contenant trois séquences ou plus avec ce type d'anomalie sera déclaré inconsistant. De plus, tout profil ayant un rapport variance/moyenne supérieur à 0,25 pour la durée de la séquence sera également éliminé. Ces heuristiques ont été déterminées en examinant les profils de la base de référence et les performances qui leur sont associées.

Ces heuristiques ont permis d'écarter 81 profils soient 4% des profils de notre base (nous rappelons que chaque utilisateur a un profil pour chaque succession de dix séquences). Ces profils étaient concentrés sur 5 utilisateurs uniquement.

Dans le chapitre 2, nos proposons une seconde méthode qui réalise une classification des profils de façon à distinguer les profils inconsistants des autres.

Pour cela les profils de la base de référence qui ont obtenu une somme TFA+TFR supérieur à 0,6 sont déclarés inconsistants. Cela, représentent 9% des profils de la base.

Avec cette méthode, la base de référence sert à entraîner un classificateur pour séparer les profils inconsistants des autres. Nous avons choisi d'utiliser les SVM comme classificateur et obtenons les résultats présentés sur le Tableau 22. Ce tableau représente la matrice de confusion du classificateur.

Tableau 22 : Matrice de confusion de la classification des profils inconsistants

Profil	Observé consistant	Observé inconsistant	Total
Prévu consistant	1243	2	1245
Prévu inconsistant	442	131	375
Total	1685	133	1818

Le taux de mal classés avec ce classificateur est de 24%. Quasiment tous les profils inconsistants sont correctement détectés, mais une grande partie des bons profils est également classée dans les profils inconsistants.

Ce résultat pose la question de la gestion des profils inconsistants. En effet au regard du Tableau 23, nous constatons une amélioration considérable des performances (significatif à 92%) mais au prix de l'élimination de 6 utilisateurs sur les 40 initiaux. Ce procédé génère de plus des problèmes importants pour 5 autres utilisateurs pour qui une grande partie des profils sont rejetés, et le système ne fonctionne qu'après plus de dix authentifications réussies après l'enregistrement grâce à la mise à jour.

Tableau 23 : Performances avant et après l'élimination des profils inconsistants

	TEE'	TFR	TFA	TFR Maximum	TFA maximum	Intervalles de confiance à 95% Pour le TEE'
Performance sur toute la population	4,9%	5,1%	4,7%	28,3%	48,6%	[2,6 ; 7,2]%
Sur les profils consistants	2,9%	3,2%	2,7%	22%	21%	[1,7 ; 4,1]%
En assouplissant le seuil pour les profils inconsistants	9,8%	2,5%	17,0%	22,0%	85,0%	[7 ; 12,6]

En situation réelle, il semble impossible de rejeter ainsi plus de 10% des utilisateurs ou même de leur imposer de refaire un enregistrement. L'assouplissement du seuil de sécurité (en le multipliant par 1,5 par exemple pour les profils inconsistants), provoque une nette amélioration des performances pour les

utilisateurs inconsistants au niveau du TFR. Par contre le TFA du système augmente pour beaucoup d'utilisateurs qu'ils aient des profils consistants ou non.

Nous manquons hélas de données (notamment pour les utilisateurs inconsistants) pour déterminer si leur profil se stabilise au cours du temps. Dans l'état actuel, nous ne sommes donc pas capables de préconiser une solution pour résoudre ce problème dans le cadre de notre application (dynamique de frappe). Néanmoins, la mise jour continue du profil permet, comme nous l'avons vu d'en atténuer les effets.

3.4. Bilan de l'étude de la dynamique de frappe

Nous avons testé l'architecture proposée dans le chapitre 2 sur un système d'authentification basé sur l'analyse de la dynamique de frappe. Les performances du système ont été évaluées sur une base de données de 40 utilisateurs sur une longue période (3 ans).

Ces travaux ont permis de valider notre proposition d'architecture en montrant que la personnalisation des paramètres du système permet un gain de performances considérable que ce soit pour les taux d'erreur ou pour le comportement du système sur le long terme. Le groupe CapMonétique a considéré les performances obtenues (Tableau 24) satisfaisantes pour renforcer la sécurité d'un système informatique à l'aide d'un système léger ne demandant ni effort supplémentaire (hors enregistrement) à l'utilisateur, ni achat de matériel supplémentaire. Cette architecture aboutit à la mise en place en situations réelles d'un logiciel d'authentification par la dynamique de frappe. Ce logiciel remplace l'interface d'ouverture de sessions de Windows (MsGina) en rajoutant une reconnaissance de la dynamique de frappe.

Des problèmes peuvent survenir en cas de modification brutale du comportement d'un utilisateur. L'intervention d'un administrateur du système est alors indispensable. Le principal problème restant à traiter concerne la détection et la gestion des utilisateurs inconsistants.

Tableau 24 : Evaluation de notre système d'authentification basé sur la dynamique de frappe

Indicateur	Performances
Coûts du système	0€ pour le matériel. Le système est très bon marché
Acceptation par l'utilisateur	Identique à l'utilisation du couple identifiant/mot de passe classique
Coûts de reconnaissance	2 séquences de touches de plus de six caractères (identifiant et mot de passe)
Coûts d'enregistrement	10 séquences (identifiant et mot de passe)
Taux d'erreur	TEE de 4,9%, ce taux est suffisant pour la plupart des applications, mais reste inférieur aux méthodes biométriques physiques
Taux d'échec à l'enregistrement	Proche de zéro, tous les utilisateurs parviennent à utiliser le système. Le seul problème qui peut survenir est une blessure sur une main qui modifie considérablement la frappe.
Contre indication	Notre système est utilisable à tous les endroits où l'on veut contrôler l'accès à des ressources informatiques (présence d'un clavier). L'utilisateur doit par contre être concentré et disponible lors de l'authentification.

Chapitre 4.
Application à la reconnaissance de signatures manuscrites

L’application de nos propositions sur un système d’analyse de la dynamique de frappe a été concluante. Cependant, pour compléter et valider complètement notre architecture, nous avons décidé de l’appliquer à un autre problème : celui de la reconnaissance de la signature manuscrite en ligne.

La signature est traditionnellement utilisée depuis de nombreuses années pour certifier des documents (contrats, chèques…). Nous proposons ici de nous intéresser à l’étude de signatures en ligne c'est-à-dire, qu’en plus d’utiliser la forme de la signature, nous disposons d’informations temporelles.

Le principe de la signature en ligne est de faire signer un utilisateur sur un dispositif d’acquisition comme une tablette graphique ou un PDA à l’aide d’un stylo optique. Cette méthode est plus fiable que la signature hors ligne (sur papier), puisque des informations sur la dynamique de la signature s’ajoutent à la forme de la signature. Néanmoins, si on parvient actuellement à écarter assez facilement les faux grossiers (quand le faussaire ne connaît pas la signature à imiter), on arrive à des taux d’erreur assez importants quand le faussaire connaît la forme à imiter ou mieux encore quand il a vu signer l’utilisateur.

Nos objectifs sont, à partir des résultats obtenus par Mathieu Wirotius au cours de sa thèse [Wirotius, 2005] d’appliquer les principes que nous proposons pour améliorer les performances du système.

Le travail effectué ne portera donc pas sur le choix des composants du système mais sur son architecture ainsi que sur ses capacités d’adaptation aux comportements des scripteurs. Nous commençons par résumer brièvement le travail de Mathieu Wirotius puis nous montrerons l’apport de nos propositions sur les performances du système de reconnaissance de signatures manuscrites produit.

4.1. Représentation des signatures et base de test

La reconnaissance de la signature manuscrite est un domaine de la biométrie actuellement très étudié. Pour plus d'informations sur les différentes méthodes utilisées pour résoudre ce problème le lecteur pourra se reporter à [Gupta et Cabe, 1997], [Griess et Jain, 2000], [Jain *et al.*, 2002] et [Wirotius, 2005].

4.1.1. Données extraites

Lors de l'analyse de signatures en ligne, une signature est représentée par une liste de points horodatés. Pour chaque point de la signature, un certain nombre de caractéristiques sont disponibles :

- Date d'enregistrement du point
- Coordonnées en X du point
- Coordonnées en Y du point
- Un booléen qui indique s'il y a pression ou non au moment de l'acquisition du point

Une signature contient entre 100 et 300 points suivant sa longueur et la fréquence d'échantillonnage du dispositif d'acquisition utilisé (TabletPC ou PDA dans notre cas).

4.1.2. Normalisation

Une fois les signatures acquises, Mathieu Wirotius propose d'effectuer une normalisation des données en trois étapes :

1. Détermination de la direction de l'axe principal d'inertie de la signature, c'est-à-dire la pente de la droite des moindres carrés du nuage de points formant la signature. Sur l'exemple de la Figure 33, l'axe d'inertie est en rouge
2. Rotation de la signature de façon à ce que l'axe d'inertie devienne horizontal (Figure 34)
3. Réalisation d'une homothétie de manière à ce que toutes les signatures soient contenues dans un rectangle de même largeur, fixée à 300 pixels.

L'homothétie consiste tout simplement à appliquer un coefficient multiplicateur ou réducteur, aux dimensions réelles, suivant les besoins. Le centre de l'homothétie est le centre de gravité de la signature. Le fait de ne pas imposer la hauteur du rectangle englobant permet de conserver les proportions de la signature

4. Une translation est ensuite effectuée sur la signature de manière à ce que son centre de gravité soit confondu avec l'origine du repère durant toute la suite du traitement (Figure 35)

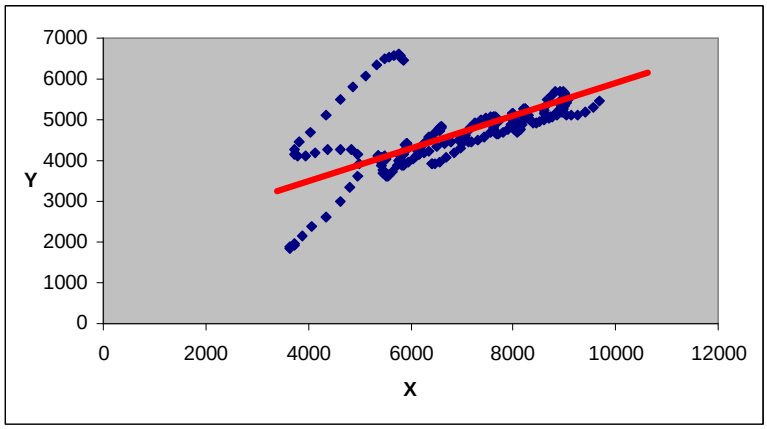


Figure 33. Axe d'inertie de la signature.

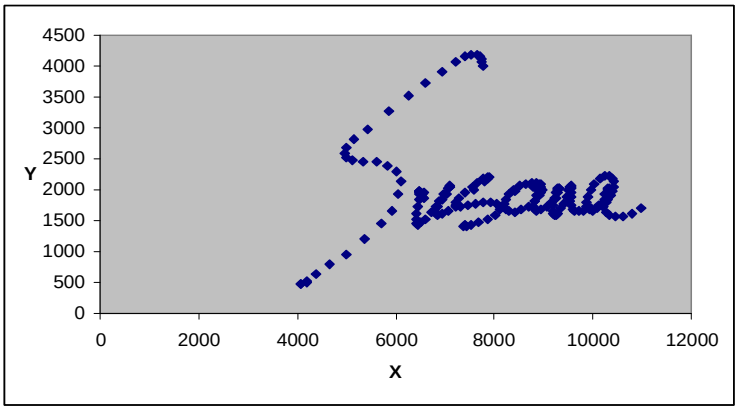


Figure 34. Signature redressée.

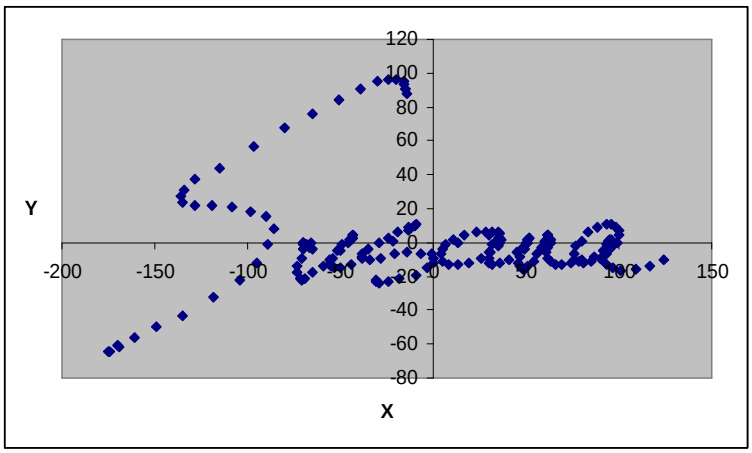


Figure 35. Translation pour positionner le centre de gravité à l'origine du repère.

Cette normalisation a pour principal but de gommer les différences que l'on peut trouver chez un même individu à cause, par exemple, d'un mauvais positionnement du support d'acquisition.

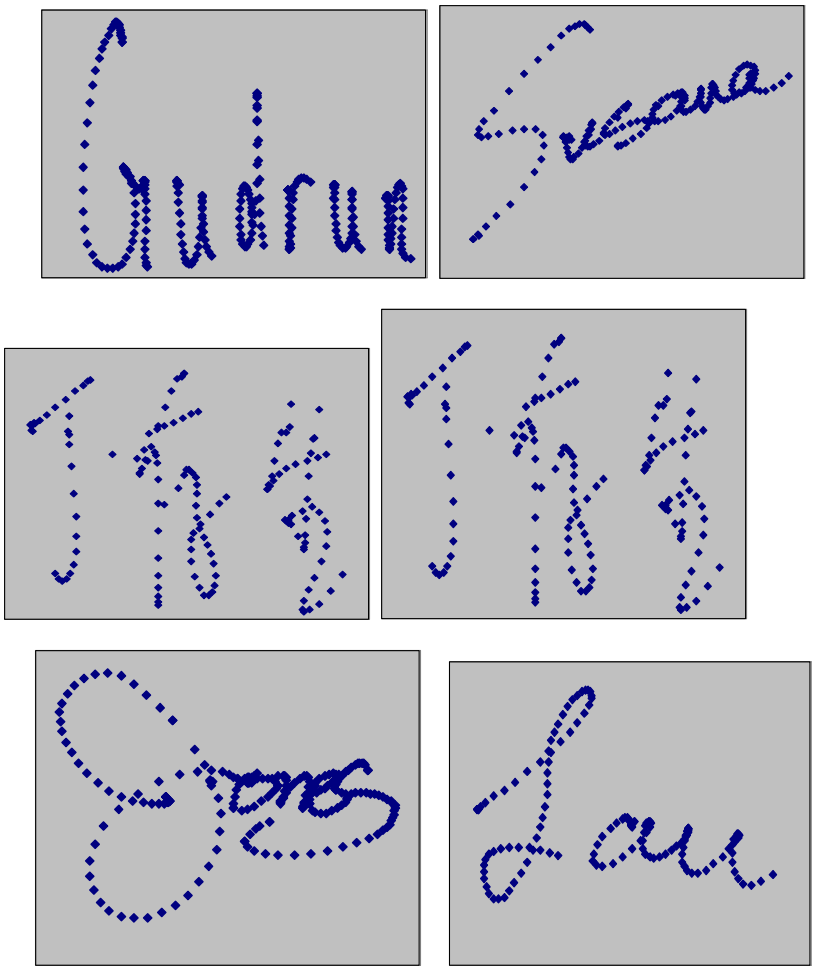


Figure 36. Exemples de signatures de la base SVC.

4.1.3. Bases utilisées

Nos travaux utilisent une base de signatures en ligne constituée pour une compétition internationale : "Signature Verification Competition" (SVC) [SVC, 2004] ayant eu lieu en 2004 dans le cadre de la conférence "International Conference on Biometric Authentication" (ICBA).

Cette base contient les signatures de 40 personnes. Pour chaque personne, on dispose de 20 signatures authentiques et de 20 faux expérimentés (Figure 36). Les faux expérimentés ont été réalisés par des personnes ayant accès à une vidéo de la personne en train de signer.

La particularité de cette base est que les signatures enregistrées ne sont pas de véritables signatures mais des signatures créées uniquement pour l'élaboration de la base. La principale conséquence de ce choix est une plus faible stabilité de la signature au cours du temps par rapport aux cas réels.

Nous nous servirons néanmoins de cette base pour construire nos bases de référence et de test. Comme pour la dynamique de frappe, nous ne disposons pas d'assez de données pour constituer des bases de développement et d'évaluation complètement indépendantes (impliquant à nouveau l'utilisation de *leave one out*).

4.2. Classificateurs utilisés

Pour authentifier un individu, Mathieu Wirotius a utilisé une architecture *Coarse to fine* dans le but d'éliminer les faux grossiers avant de procéder à un examen plus fin des signatures plus semblables. Dans nos travaux, nous avons éliminé l'étape *Coarse*, car si elle permet d'éliminer facilement les faux aléatoires, elle est inefficace pour détecter les faux expérimentés. Nous avons donc préféré nous concentrer sur l'étape *fine* et améliorer celle-ci à l'aide de nos propositions.

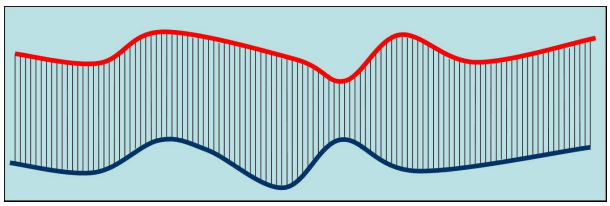
Mathieu Wirotius a choisi d'utiliser une mesure de similarité appelé Distance élastique ou *Dynamic Time Warping* DTW([Plamondon et Parizeau, 1988] et [Hastie *et al.*, 1992]), afin d'authentifier les signatures. Nous allons brièvement présenter cet outil et les améliorations ayant permis à M. Wirotius [Wirotius *et al.*, 2004] d'obtenir plusieurs classificateurs.

4.2.1. Distance élastique ou Dynamic Time Warping (DTW)

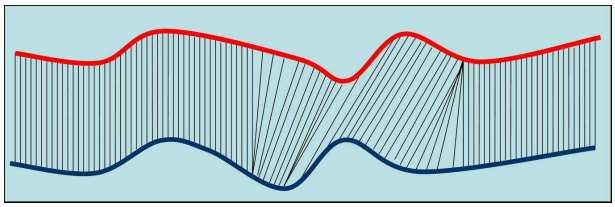
L'avantage du DTW est de prendre en compte les différences de rythme et les décalages temporels. Pour cela, la distance DTW réalise une mise en correspondance point à point entre deux signaux avec l'avantage d'être insensible aux différences de longueur (Figure 37).

Les points initiaux de chacune des deux courbes *C* et *C'* sont mis en correspondance. Soient Pt_i le ième point de la courbe *C* et Pt'_j le jème point de la courbe *C'*, les points Pt_i et Pt'_j ayant été mis en correspondance (Figure 38). Soient d_1, d_2 et d_3 les distances suivantes :

$$\begin{cases} d_1 = Dist(Pt_{i+1}, Pt'_j) \\ d_2 = Dist(Pt_i, Pt'_{j+1}) \\ d_3 = Dist(Pt_{i+1}, Pt'_{j+1}) \end{cases}$$



(a)



(b)

Figure 37. Mise en correspondance point à point entre deux signatures sans prise en compte des décalages temporels (a) et en appliquant l'algorithme DTW (b).

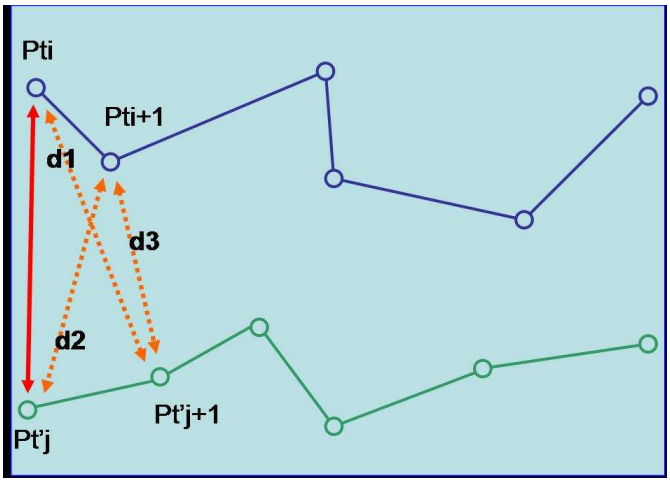


Figure 38. Principe de l'algorithme DTW.

Les points Pt_k et Pt'_l sont mis en correspondance de telle sorte que :

$$Dist(Pt_k, Pt'_l) = \min(d_1, d_2, d_3)$$

Classiquement, la métrique utilisée est la distance spatiale euclidienne entre les points des signatures.

Une fois la mise en correspondance des points effectuée, on peut calculer la distance entre les deux courbes en effectuant la somme des distances entre couples de points associés. DTW autorise deux transformations suivant l'axe temporel lors de l'appariement de deux points d'une signature :

- l'allongement
- le rétrécissement

Cette méthode de comparaison de courbes est très utilisée dans le domaine de l'authentification par signature manuscrite car elle permet une plus grande flexibilité et n'est pas perturbée par de légères variations.

4.2.2. Amélioration du DTW et création des classificateurs

4.2.2.1. Amélioration du DTW

Mathieu Wirotius a proposé plusieurs améliorations du DTW. La première amélioration propose de réduire le nombre de points représentatifs des signatures en conservant uniquement des points de vitesse minimum afin de faciliter le travail de l'algorithme d'appariement [Wirotius *et al.*, 2004; Wirotius *et al.*, 2004]. Cette méthode, comparable à un lissage, permet d'améliorer les performances par rapport à l'utilisation de l'ensemble des points initiaux lors du DTW.

Pour apparier deux points de la signature, la plupart du temps, seules les coordonnées spatiales sont utilisées. Le calcul de dissimilarité entre deux signatures est ensuite calculé comme décrit dans l'équation (23) .

Soient *S1* et *S2* deux signatures, et *nbCorresp* le nombre de points de *S1* ayant des correspondances dans *S2*. On note Pt_i un point quelconque de *S1* et P'_i l'ensemble des points correspondants dans *S2*. La distance classique entre les deux signatures *S1* et *S2* est donnée par l'équation (23).

$$dist(S1,S2) = \sum_{i=1}^{nbCorresp} \sum_{P' \in P'_i} DistPt(Pt_i, P') \qquad \textbf{(23)}$$

Une autre amélioration proposée par Mathieu Wirotius est de ne prendre en compte qu'un seul appariement par point afin d'éviter le cumul d'erreur. Nous précisons $P_{t'i}=\{Pt'_{i1}, Pt'_{i2}, \ldots\}$ l'ensemble des points appariés avec *Pt'i* indicés suivant les temps croissants. Ainsi la formule de distance est modifiée pour obtenir l'équation (24) .

$$dist(S1,S2) = \sum_{i=1}^{nbCorresp} DistPt(Pt_i, Pt'_{i,1}) \qquad \textbf{(24)}$$

Cette distance est normalisée en la divisant par le nombre de correspondances afin d'être indépendante du nombre de points de la signature. La formule finale (et utilisée par la suite) suit donc l'équation (25).

$$Dist(S1,S2) = \frac{1}{nbCorresp} dist(S1,S2) \quad \textbf{(25)}$$

Ce résultat peut être vu comme la moyenne des distances entre points mis en correspondance dans les deux signatures.

Enfin, la dernière amélioration apportée à DTW vise à prendre en compte la variabilité propre à chaque signataire. Elle consiste à soustraire à la mesure obtenue la plus petite distance intra apprentissage (équation (26)) *minDistApp* calculée entre les signatures d'apprentissage Sa_i contenues dans le profil de l'utilisateur (équation (27)).

$$\boldsymbol{minDistApp = \min_{i,j,i\neq j} Dist(Sa_i, Sa_j)} \quad \textbf{(26)}$$

$$\boldsymbol{DistVar(S1,S2) = Dist(S1,S2) - minDistApp} \quad \textbf{(27)}$$

4.2.2.2. Création de classificateurs

Pour améliorer encore les performances, Mathieu Wirotius propose d'utiliser trois mesures différentes pour comparer des signatures en ligne :

- Distance spatiale

L'algorithme DTW amélioré peut être appliqué sur les coordonnées (x,y) de chaque point des signatures. Le classificateur ainsi obtenu effectue une reconnaissance de signature hors ligne puisque seules les informations spatiales sont utilisées lors du calcul de la distance entre deux points (distance euclidienne simple).

- Distance temporelle

Pour prendre en considération l'aspect temporel global de la signature, au lieu de calculer une distance euclidienne entre les coordonnées des points, le temps écoulé entre le premier point et chaque point considéré peut remplacer les coordonnées spatiales. Une normalisation linéaire du temps sur l'intervalle [0,1] est effectuée. La formule pour le calcul de la distance entre deux points Pt et Pt' est donnée par l'équation (28).

$$\boldsymbol{DistPt_{temporelle}(Pt,Pt') = |t(Pt) - t(Pt')|} \quad \textbf{(28)}$$

- Distance curviligne

Afin de compléter l'information apportée par la distance temporelle, une troisième mesure de dissimilarité a été mise en place, permettant de prendre en compte « l'efficacité » du tracé. Le principe de calcul consiste à comparer la distance parcourue depuis le premier point d'acquisition jusqu'au point considéré.

Une normalisation linéaire de la longueur sur l'intervalle [0,1] est effectuée pour pouvoir comparer des signatures n'ayant pas des longueurs identiques. La distance entre les signatures est calculée suivant les équations (29) et (30).

$$\boldsymbol{Dist(S1,S2)=\sum_{i=1}^{n}\left|\frac{Longueur(Pt_i)}{Longueur(S1)}-\frac{Longueur(Pt'_i)}{Longueur(S2)}\right|} \quad \textbf{(29)}$$

$$Longueur(Pt_i)=\sum_{j=2}^{i} Dist(Pt_{j-1},Pt_j) \quad \textbf{(30)}$$

Trois classificateurs fournissant trois scores (mesures de dissimilarité) sont ainsi disponibles et peuvent être fusionnés pour produire la décision finale.

4.3. Etape de fusion

4.3.1. Principe

Les trois distances spatiale, temporelle et curviligne utilisant des informations différentes, une fusion de ces trois informations peut donc être envisagée. Cette fusion est réalisée avec l'opérateur *Somme* et en affectant à chacune d'entre elles un poids de manière similaire à ce qui avait été proposé pour la dynamique de frappe au clavier. Le score est obtenu à l'aide de l'équation (31).

$$d = \alpha \times distS + \beta \times distT + \gamma \times distL \qquad \textbf{(31)}$$

Avec les contraintes sur les poids suivantes :

$$\begin{cases} \alpha + \beta + \gamma = 1 \\ \alpha, \beta, \gamma \in [0,1] \end{cases}$$

Dans un premier temps, (correspondant à la proposition finale de M. Wirotius) les poids peuvent être fixés de façon globale c'est-à-dire identiques pour tous les utilisateurs et de façon à minimiser le TEE en explorant le cube de variation des paramètres systématiquement avec un pas de 0,1.

4.3.2. Performances

Comme nous l'avons déjà précisé, le calcul des performances a été fait sur la base SVC qui contient 40 utilisateurs. Pour chaque individu, le profil est composé par les cinq premières signatures, le TFR est calculé sur les quinze signatures restantes. Le TFA est calculé pour les faux aléatoires sur l'ensemble des signatures n'appartenant pas à l'individu, hors faux entraînés (1560 signatures). Pour les faux expérimentés, le TFA est calculé sur vingt signatures d'imposteurs entraînés. Le Tableau 25 présente les taux obtenus lorsque le poids affecté à chaque classificateur est fixé empiriquement à 0,33.

Tableau 25 : Performance des méthodes avec les faux entraînés (α=β=γ=0,33)

DTW	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
spatial	32%	60%	[27 ; 37]%
temporel	25%	47%	[21,5 ; 28,5]%
curviligne	36%	75%	[32 ; 40]%
Fusion	25%	60%	[21 ; 29]%

Le Tableau 25 présente les résultats obtenus sur des faux entraînés. Les taux d'erreur sont très importants et dépassent pour certains individus les 50%. La fusion n'améliore pas les résultats par rapports à la meilleure méthode. La conclusion que l'on peut tirer de ce tableau est, qu'en l'état, le système ne fait pas beaucoup mieux que le hasard et même pire pour certains utilisateurs. Cela peut s'expliquer par deux facteurs :

- Le problème est difficile
- Le seuil de décision n'est pas adapté

Le problème de détection de faux expérimentés est certes difficile, surtout pour la base SVC dans laquelle les utilisateurs n'ont eu que peu de temps pour apprendre leur signature ; mais cela n'explique pas tout. L'examen de la base, utilisateur par utilisateur, montre que beaucoup d'entre eux obtiennent avec le seuil global un TFR ou un TFA égal à 1 avec le second taux très inférieur et parfois proche de zéro. Ce fait montre que le seuil de décision est manifestement inadapté. Il doit donc être possible d'améliorer considérablement les résultats par une personnalisation des paramètres comme nous le préconisons.

Tableau 26 : Performance des méthodes sur les faux aléatoires (α=β=γ=0,33)

DTW	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
spatial	6,3%	43,0%	[3,8 ; 8,8]%
temporel	5,0%	16,0%	[3,7 ; 6,3]%
curviligne	9,0%	50,0%	[5,8 ; 12,2]%
Fusion	3,1%	34,0%	[2,1 ; 4,1]%

Quand on travaille avec des faux aléatoires (Tableau 26) les résultats obtenus sont biens meilleurs et correspondent à ce que l'on obtenait dans le cadre de la dynamique de frappe. La fusion améliore considérablement les résultats par rapports à la meilleure méthode qui est ici le DTW temporel (significatif avec un risque de première espèce inférieur à 1%). Il reste intéressant de voir si une personnalisation des paramètres permet d'améliorer ces taux.

4.4. Personnalisation du système

4.4.1. Intérêt d'une personnalisation

Les tests préliminaires ont montré surtout dans le cadre de la détection de faux entraînés, la nécessité très forte d'individualisation des paramètres du système. Pour la base SVC, comme le nombre de signatures disponibles pour chaque utilisateur est faible et que celles-ci ont été acquises sur une période inconnue (mais qui semble être assez courte), nous avons renoncé à effectuer un travail sur la mise à jour du profil des utilisateurs. Nous nous sommes donc uniquement concentrés sur la personnalisation des paramètres, c'est-à-dire l'adaptation du seuil de décision et les des poids de fusion au comportement de chaque individu.

Tableau 27 : Apport de la personnalisation des paramètres sur la détection des faux entraînés

DTW	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
paramètres égaux	25,0%	60,0%	[21 ; 29]%
paramètres optimaux globaux	23,0%	50,0%	[20 ; 26]%
paramètres optimaux individuels	7,0%	21,5%	[4 ; 10]%

Nous pouvons constater (Tableau 27) que, pour la détection de faux entraînés, l'utilisation de poids de fusion différents pour chaque méthode n'améliore que peu les résultats, par contre même si les taux d'erreur restent élevés (7%), l'utilisation de paramètres optimaux pour chaque utilisateur, calculés à l'aide des signatures d'imposteurs pour minimiser la somme TFA plus TFR, entraîne un gain de

performance énorme (significatif avec un risque de première espèce inférieur à 1%). Il est donc judicieux de chercher à fixer localement les paramètres.

Tableau 28 : Apport de la personnalisation des paramètres sur les faux aléatoires

DTW	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres égaux	3,10%	44,00%	[2,1 ; 4,1]%
Paramètres optimaux globaux	2,70%	13,00%	[2 ; 3,4]%
paramètres optimaux individuels	0,06%	1,50%	[0.05 ; 0,07]

Le gain de performance sur les faux aléatoires et l'utilisation de paramètres optimaux individuels est encore une fois très important (significatif avec un risque de première espèce inférieur à 1%) on approche les 0% d'erreur pour le TEE ! L'utilisation de paramètres locaux doit donc absolument être envisagée.

4.4.2. Estimation des paramètres

Il est donc nécessaire de déterminer la technique d'obtention des paramètres individuels pour chaque utilisateur. Il s'agit de déterminer les données (base de référence) et caractéristiques (estimateurs) à utiliser pour déterminer automatiquement les valeurs des paramètres. Pour cela, nous allons reprendre la démarche d'estimation que nous avons déjà suivie pour la dynamique de frappe.

4.4.3. Construction du vecteur *Ref*

Nous construisons le vecteur *Ref*, d'un profil en extrayant des informations sur les cinq signatures d'apprentissage disponibles, ce vecteur regroupe donc :

- La moyenne du DTW spatial
- La variance du DTW spatial
- La moyenne du DTW temporel
- La variance du DTW temporel
- La moyenne du DTW curviligne

- La variance du DTW curviligne
- La longueur totale moyenne de la signature
- La variance de la longueur totale de la signature
- La moyenne du temps total d'exécution de la signature
- La variance du temps total d'exécution de la signature

Initialement, nous avions extrait beaucoup d'autres caractéristiques mais leurs corrélations trop faibles avec les paramètres nous ont conduits à les rejeter pour ne pas perturber les analyses ultérieures.

4.4.4. Estimateurs utilisés

Concernant les estimateurs, nous avons décidé de reprendre les méthodes que nous avions déjà testées sur la dynamique de frappe.

La difficulté principale rencontrée dans le cadre de la vérification de signatures est le manque de profils disponibles pour construire la base de référence. En effet, dans le cadre de la dynamique de frappe, si nous avions juste un peu plus d'utilisateurs (42 contre 40 pour la signature), nous avions pu isoler pour chaque utilisateur plusieurs profils avec des jeux de paramètres différents, correspondant à l'évolution du comportement de l'utilisateur au cours du temps. On disposait donc, pour déterminer le jeu de paramètres d'un profil, d'une grande quantité d'autres profils et de leurs jeux de paramètres associés. Ici pour un utilisateur, seuls les profils et les paramètres optimaux des 39 autres utilisateurs restants sont disponibles.

Comme pour la dynamique de frappe, l'affectation directe des paramètres à partir des caractéristiques extraites des profils, n'a pas donné de résultats concluants (les taux d'erreur étaient largement plus élevés qu'avec des paramètres globaux). La raison de ces taux d'erreur très importants vient d'une inadéquation entre les poids et le seuil de décision qui entraîne des taux d'erreur de type (1,0) pour le TFA et le TFR.

Les méthodes de création de classes de comportement d'utilisateurs avaient été une bonne alternative pour la dynamique de frappe, nous avons donc également testé cette approche sur les signatures.

A partir de la base de référence, 4 classes de comportement ont été crées à partir des vecteurs *Ref*. Le choix de quatre classes a été fait avec l'objectif de créer le plus possible de classes avec un nombre suffisant d'utilisateurs, nos tests ont montrés que quatre était le nombre idéal de classes sur la base considérée. La création des classes a été effectuée à l'aide de la méthode des k-means. Un utilisateur est affecté à la classe dont le centre de gravité est le plus proche de lui. Nous avons ensuite calculé, pour chaque classe, le jeu de paramètres le plus efficace, en explorant de façon exhaustive l'espace des possibilités.

Les paramètres obtenus sont présentés dans le Tableau 29. Les quatre classes ont des jeux de poids très voisins pour les faux aléatoires, seul le seuil de décision

diffère vraiment. Les classes ont un degré de permissivité très différent avec un seuil variant de -0,12 à 0,5.

L'examen des paramètres obtenus pour les faux entraînés, montre que les poids affectés au DTW temporel sont très élevés pour deux classes, et moyen pour les deux autres, indiquant que les caractéristiques permettant d'écarter les imposteurs diffèrent suivant les individus.

Tableau 29 : Paramètres affectés à chaque classe

	Faux aléatoires			Faux entraînés		
Classe	alpha	bêta	seuil	alpha	bêta	seuil
1	0,2	0,8	0,11	0	0,9	-0,27
2	0,2	0,6	-0,12	0,3	0,4	-0,35
3	0,3	0,6	0,16	0,2	0,5	-0,2
4	0,4	0,5	0,5	0	0,8	-0,16

Tableau 30 : Estimation des paramètres sur les faux entraînés

Méthode	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres optimaux	7,0%	21,5%	[4 ; 10]%
Paramètres globaux	23,0%	50,0%	[20 ; 26]%
Paramètres estimés	20,0%	41,0%	[17-23]%

Tableau 31 : Estimation des paramètres sur les faux aléatoires

Méthode	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres optimaux	0,06%	1,5%	[0.05 ; 0,07]%
Paramètres globaux	2,7%	13%	[2 ; 3,4]%
Paramètres estimés	1,0%	7,3%	[0,5 ; 1,5]%

Les Tableaux Tableau 30 et Tableau 31 présentent les résultats de l'estimation des paramètres sur les faux entraînés et aléatoires. Pour les faux entraînés, les résultats sont un peu meilleurs avec les paramètres individualisés, mais

restent très proches des taux initiaux. Nous ne sommes donc pas parvenus à créer des classes consistantes ayant un jeu de paramètres commun pour l'ensemble des utilisateurs.

Pour les faux aléatoires, l'amélioration est par contre très importante et on atteint 1% de TEE (significatif avec un risque de première espèce inférieur à 1%). Ces résultats sont très encourageants.

Nous avons également essayé de créer des classes de paramètres plutôt que des classes de comportement. Il faut alors produire des classificateurs permettant de déterminer à quelle classe de paramètres appartient un nouvel utilisateur. Quatre classes ressortent de notre analyse. Ces classes sont présentées sur la Figure 39 et ont été crées à l'aide des k-means sur l'espace des paramètres. Les résultats sont présentés à l'aide d'une ACP afin de visualiser les classes en deux dimensions. Les paramètres affectés à chaque classe sont la moyenne des paramètres de tous les individus de la classe. Les paramètres moyens affectés à chaque classe sont présentés dans le Tableau 32. Trois des classes crées B, C et D sont des classes assez moyennes, la classe A regroupe des utilisateurs qui nécessitent des paramètres plus extrêmes. L'autre enseignement apporté par la Figure 39 est la présence d'individus extérieurs au cercle unité. Du fait, de leur éloignement, ces individus, sont affectés à des classes assez distantes et risquent de poser problème dans la suite.

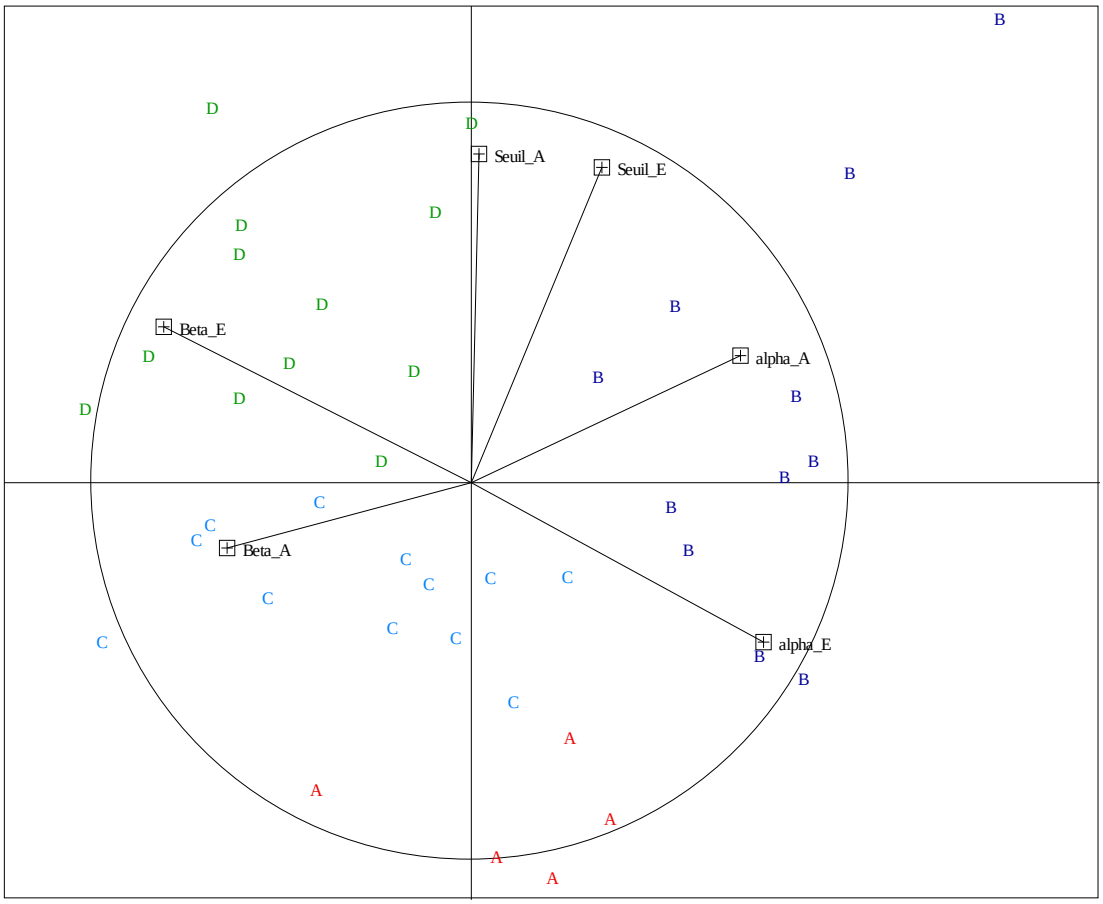


Figure 39: Création de classes de paramètres

Les paramètres des classes de paramètres sont très différents de ceux issus des classes de comportement, ils se différencient autant par leur seuil de sécurité, que par le poids affecté à chaque méthode. Ceci s'explique par les différences de la méthode de création des classes, les classes de paramètres sont beaucoup plus différenciés car plus proches des paramètres locaux optimaux, alors que les classes crées sur les caractéristiques dont des paramètres plus moyen.

Tableau 32 : Paramètres moyens des classes de paramètres

Classe	Faux entraînés			Faux aléatoires		
	alpha	beta	seuil	Alpha	beta	seuil
1	0,7	0,1	-0,6	0,25	0,5	-0,35
2	0,6	0,2	-0,2	0,6	0,1	-0,1
3	0,2	0,5	-0,4	0,1	0,2	-0,35
4	0,1	0,7	-0,25	0,2	0,4	0,1

Nous avons d'abord essayé les k-ppv comme classificateurs des utilisateurs pour leur affecter une classe de paramètres puis les SVM. Nous avons constaté que les performances des deux classificateurs sont du même ordre (Tableau 33).

Tableau 33 : Estimation des paramètres par clustering puis classification

	Faux entraînés			Faux aléatoires		
Méthode	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'	TEE moyen	TEE maximum	Intervalles de confiance à 95% Pour le TEE'
Paramètres globaux	23,0%	50,0%	[20 ; 26]%	2,7%	13,0%	[2 ; 3,4]%
Paramètres optimaux	7,0%	21,5%	[4 ; 10]%	0,06%	1,5%	[0.05 ; 0,07]%
Paramètres estimés k-ppv	23,0%	46,0%	[20 ; 26]%	2,1%	11,0%	[1,4 ; 2,8]%
Paramètres estimés SVM	24,0%	46,0%	[21 ; 27]%	1.9%	12,0%	[1,1 ; 2,7]%

Les performances sont décevantes par rapport à la dynamique de frappe. Les améliorations par rapport à un jeu de paramètres globaux sont inexistantes. Nous n'avons pas gagné globalement, mais nous avons néanmoins gagné un peu de performance pour les utilisateurs extrêmes. Les résultats sont par contre bien meilleurs pour les faux aléatoires.

Nous avons cherché à expliquer la faiblesse de ces résultats en examinant les classes de paramètres dans l'espace du vecteur *Ref*. Pour présenter ces résultats en deux dimensions, nous avons réalisé une ACP (Figure 40). On remarque que les classes sont entremêlées. Ainsi, quel que soit le classificateur, il sera difficile de parvenir à une bonne séparation des classes. De plus, les utilisateurs se trouvant loin du cercle semblent également très difficiles à classer.

Nous pensons que le problème peut provenir des caractéristiques que nous avons choisies pour estimer les paramètres. L'espace de travail ne semble pas être le bon. Nos essais de sélection de caractéristiques, par un algorithme de type SFFS, ont également échoué. Le travail devrait donc s'orienter vers la recherche de nouvelles caractéristiques permettant de mieux déterminer les paramètres.

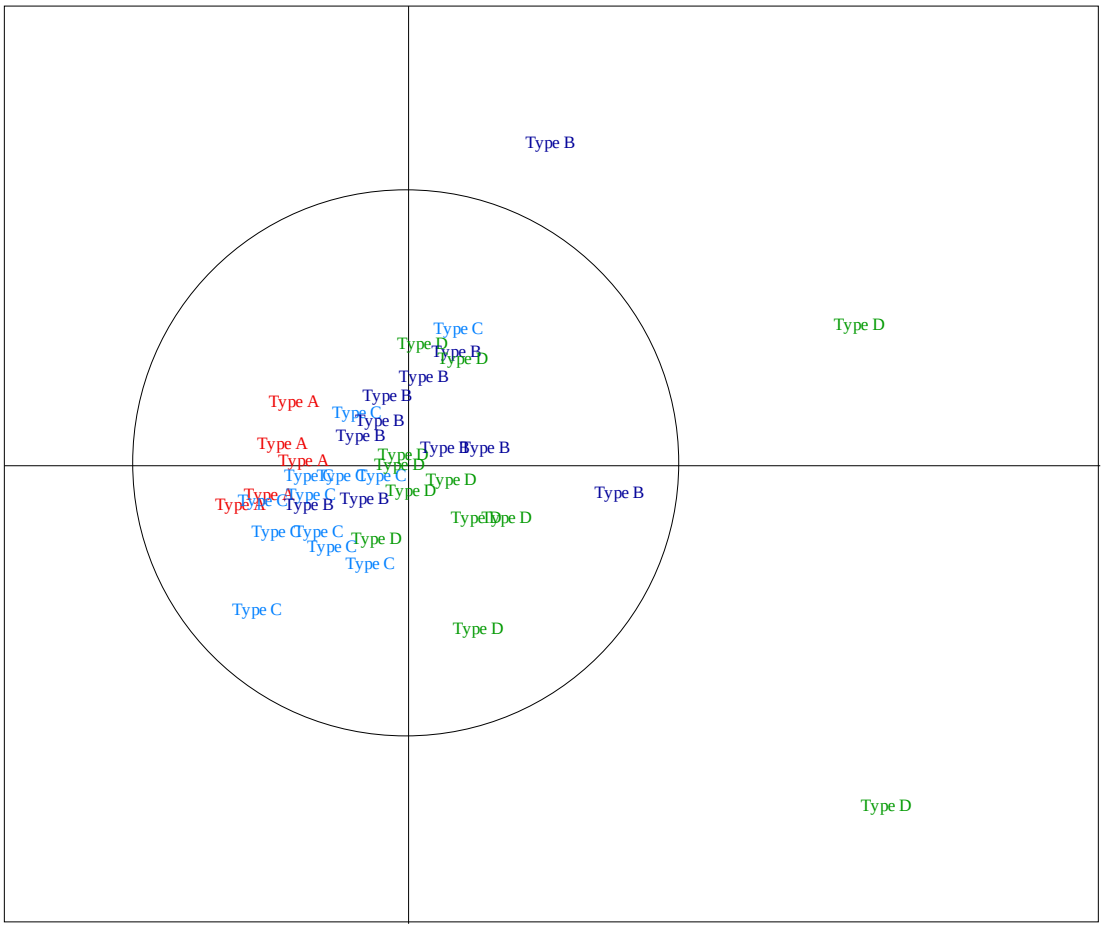


Figure 40 : Classes de paramètres dans l'espace du vecteur *Ref*

4.5. Bilan sur la signature

Nos travaux sur la signature ont donné des résultats plus mitigés que ceux obtenus sur la dynamique de frappe. Les problèmes de stabilité et de significativité des résultats (intervalles de confiance) rencontrés lors des évaluations statistiques sont bien moins présent que pour l'étude de la dynamique de frappe. Nous avons montré que l'architecture proposée pouvait se transposer aussi bien sur l'un que sur l'autre des problèmes. L'utilisation de paramètres individuels améliore les performances de façon spectaculaire dans les deux cas. Mais, pour la signature manuscrite, les estimateurs de paramètres que nous avons proposés donnent des résultats moins bons que ceux attendus. Cela vient sans doute, en partie, de la nature un peu artificielle de la base SVC. Nous pensons néanmoins, qu'en continuant à travailler sur cette application nous parviendrons à déterminer un ensemble de caractéristiques qui permettra d'estimer de façon correcte les paramètres utiles au moteur d'authentification.

Conclusion et perspectives

Cette thèse réalisée en partenariat avec le groupe CapMonétique, avait pour objectif la mise en place d'un système d'authentification biométrique léger et fiable. Cette problématique nous a conduits à étudier différents systèmes biométriques, et à dégager les problèmes et solutions qui sont communs à tous. Parmi ces systèmes biométriques, ceux utilisant des techniques basées sur la biométrie comportementale apparaissent comme très prometteurs de part leur facilité d'utilisation, leur bonne acceptation par les utilisateurs et leur faible coût. Mais, ces méthodes ont des contraintes particulières :

- évolution des caractéristiques biométriques comportementales au cours du temps
- très grande variabilité de comportement suivant les utilisateurs
- pas de données d'imposteurs disponibles pour l'apprentissage

Ces contraintes modifient considérablement la façon dont doit être envisagée la mise en place de ces systèmes par rapport à l'utilisation d'un système d'authentification basée sur des caractéristiques physiques.

Nos travaux proposent différentes solutions pour réduire l'effet des contraintes citées ci-dessus. Notre principale proposition est de mettre en place des méthodes de personnalisation automatique du système de manière à l'adapter le plus possible à chaque utilisateur. Afin de pouvoir réaliser cette adaptation simplement, nous préconisons d'utiliser plusieurs classificateurs différents, dont les décisions seront fusionnés. Cette phase de fusion permet d'individualiser, les paramètres du système (le seuil de décision et les poids associés aux différents classificateurs).

L'importance de la décision renvoyée par chacun des classificateurs dépend ainsi du comportement estimé de chacun des utilisateurs. Chaque classificateur fournissant un point de vue différent sur les données biométriques acquises.

Pour pouvoir mettre en place cette proposition, nous utilisons une base de référence, c'est-à-dire une base contenant des informations biométriques acquises au cours de la phase de conception du système et cherchant à caractériser le plus

fidèlement possible le comportement biométrique des futurs utilisateurs de l'application réelle désirée. Cette base doit contenir des données d'utilisateurs et d'imposteurs. La qualité de cette base est essentielle pour garantir les performances du système.

L'estimation des paramètres individuels nécessite l'ajout d'une « couche supplémentaire » aux systèmes d'authentification et complique donc leur architecture. Nos expérimentations ont montré qu'il était préférable de créer des classes de comportement (d'utilisateurs) partageant le même profil (les même paramètres individuels) plutôt que de chercher à personnaliser à outrance le système (1 utilisateur=1 jeu de paramètre).

Un algorithme de *clustering* (k-means) classique est suffisant pour déterminer les classes d'utilisateurs (créées à partir des paramètres optimaux ou des caractéristiques biométriques des utilisateurs présents dans la base de référence). Cette complexification est néanmoins largement compensée par les gains de performances obtenus par ce type de démarche.

Le deuxième axe d'amélioration sur lequel nous avons concentré nos travaux vise à permettre l'évolution du profil de chaque utilisateur au cours du temps afin de suivre les modifications de son comportement. Ces modifications de comportement peuvent être dues à l'utilisateur lui-même ou bien à une modification de l'environnement (matériel).

En plus de ces deux propositions principales, nous préconisons tout au long de ce manuscrit, différents choix ou études à effectuer lors de la conception de systèmes biométriques. Rappelons notamment que les modes d'évaluations actuels sont souvent incomplets et réalisés dans des conditions tellement variables qu'il est impossible de les utiliser pour comparer deux systèmes. Nous espérons donc que nos recommandations sur ce point (ajout d'information complémentaires aux simples taux d'erreur classiques) soient suivies afin de faciliter les comparaisons entre méthodes. Ils seraient d'ailleurs aussi opportun de constituer et de diffuser largement plus de bases de tests (pas facile pour les donnés biométriques souvent considérées comme personnelles).

Afin de valider nos propositions et afin de répondre à la demande du groupe Capmonétique, nous avons effectué des tests sur deux applications (la dynamique de frappe et la reconnaissance de signature manuscrite). Ces tests ont montré dans les deux cas l'intérêt de nos propositions. Nous avons constaté un gain de performance

important. Nous passons de 13,5% pour le TEE' sans mise à jour et sans paramètres personnalisés à 4,9% avec nos propositions pour la dynamique de frappe, que ce soit en réalisant une mise à jour du profil ou en utilisant des paramètres personnels. Pour la signature le TEE passe de 2,7 à 1% en utilisant des paramètres personnalisés. Nous sommes parvenus à faire mieux qu'avec des paramètres communs à tous les utilisateurs mais nous restons encore loin des performances atteintes lors de l'usage des paramètres individuels optimaux.

A notre avis, ces bonnes performances peuvent encore être considérablement améliorées. Pour ce faire, les futurs travaux doivent porter sur deux axes :

1. l'amélioration de l'architecture, c'est-à-dire de la structure du système biométriques quelles que soient les données utilisées. Nous envisageons de suivre de nombreuses pistes :

- Améliorer et automatiser la recherche des caractéristiques utilisables pour personnaliser le système. Un des problèmes de notre architecture est la construction du vecteur *Ref* qui entrant en jeu dans la personnalisation du système. Il s'agit de déterminer quelles sont les caractéristiques à extraire du profil d'un utilisateur et à les utiliser pour déterminer ses paramètres individuels. Si la nature de ces caractéristiques dépend du problème, les méthodes utilisées pour sélectionner et éventuellement combiner les caractéristiques peuvent être placées dans l'architecture et être communes à tous les problèmes biométriques. Certaines caractéristiques peuvent même être extraites quel que soit le système (moyenne et écart types de scores des classificateurs par exemple).
- Mettre en place un traitement spécifique des profils reconnus comme inconsistants. Nos tests ont montré que les erreurs provenaient en grande partie d'une infime minorité des utilisateurs. Nous avons présenté une méthode pour identifier ces profils, mais si cette méthode donne des résultats prometteurs, nous ne sommes pas parvenus à mettre en place un traitement spécifique convenable pour ces utilisateurs.
- Construire un indicateur de qualité objectif pour évaluer la pertinence de la base de référence. La base de référence est à la base de tout notre système, une base de référence de qualité médiocre peut engendrer des performances catastrophiques. Ces mauvaises performances dues au manque de représentativité de la base de références risquent de plus d'apparaitre qu'en situation réelles, puisque souvent la base de test est construite de la même façon que la base de référence. Un indicateur de qualité qui reste à définir pourrait aider à évaluer cette base.

Ces pistes permettront d'améliorer l'architecture du système et donc les résultats quelles que soit les données biométriques analysées.

2. la deuxième partie du travail est l'amélioration de l'adaptation de cette architecture à chaque problème. Par exemple, pour la signature manuscrite, nous pensons qu'il est possible d'améliorer encore les performances en travaillant sur ce système lui-même. Les classificateurs utilisés dans notre application actuelle ne semblent, en effet, pas convenir pour la détection de faux expérimentés. Ce choix peut expliquer en partie les résultats constatés. De même, toujours pour la reconnaissance de signature manuscrite, la construction du vecteur *Ref* reste à notre avis à améliorer.

L'architecture proposée dans cette thèse est générique et peut aider à la mise en place de différents systèmes d'authentification biométrique. La mise en place de nos recommandations implique tout de même une réflexion spécifique importante à divers niveaux (choix des caractéristiques, des classificateurs….) dont dépendra largement le succès du système d'authentification biométrique implémenté. Si ces réflexions sont menées en respectant nos propositions, il est probable que des gains de performances importants soient obtenus.

Références

[Alpaydin *et al.*, 2000] Alpaydin E., Kaynak C., Alimigolu F. (2000). Cascading Multiple Classifiers and Representations for Optical and Pen-Based Handwritten Digit Recognition. *Seventh International Workshop on Frontiers in Handwriting Recognition*, pp. 453-462.

[Anagun et Cin, 1998] Anagun A. S., Cin I. (1998), A neural network based computer access security system for multiple users. *Computers and Industrial Engineering*, **35**, 1, pp. 351-354.

[Bellman, 1961] Bellman R. E. (1961). Adaptive Control Processes. Princeton University Press, 274 p.

[Belur, 1991] Belur V. (1991). Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society, 447 p.

[Ben-Yacoub, 1999] Ben-Yacoub S. (1999). Multi-Modal Data Fusion for Person Authentication using SVM. *Second International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, pp. 25-30.

[Benzécri, 1973] Benzécri J.-P. (1973). Analyse des données. T2 (leçons sur l'analyse factorielle et la reconnaissance des formes et travaux du Laboratoire de statistique de l'Université de Paris 6. T. 2 : l'analyse des correspondances). Dunod, 632 p.

[Bergadano *et al.*, 2002] Bergadano F., Gunetti D., Picardi C. (2002), User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, **5**, 4, pp. 367-397.

[Biopassword] Biopassword, http://www.biopassword.com.

[Bleha *et al.*, 1990] Bleha S., Slivinsky C., Hussien B. (1990), Computer-Access Security Systems Using Keystroke Dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 12, pp. 1217-1222.

[Carpenter et Grossberg, 1987] Carpenter G. A., Grossberg S. (1987), ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26**, 23, pp. 4919-4930.

[Chen et Chang, 2004] Chen W., Chang W. (2004). Applying Hidden Markov Models to Keystroke Pattern Analysis for Password Verification. *IRI*, pp. 467-474.

[Cho *et al.*, 2000] Cho S., Cand H., Han D., Kim H. (2000), Web based keystroke dynamics identity verification using neural network *Journal of Organizational Computing and Electronic Commerce*, **10**, pp. 295-307.

[CNIL] Commission nationale de l'informatique et des libertés, http://www.cnil.fr/.

[Coltell *et al.*, 1999] Coltell O., Badia J. M., Torres G. (1999). Biometric Identification System Based in Keyboard Filtering. *IEEE International Carnahan Conference on Security Technology*, Madrid,Spain, pp. 203-209.

[Dietterich, 2000] Dietterich T. G. (2000), Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science*, **1857**, pp. 1-15.

[Eriksson *et al.*, 2000] Eriksson L., Johansson E., Muller M., Wold S. (2000), On the selection of the training set in environmental QSAR analysis when compounds are clustered. *J. Chemometrics*, **14**, pp. 599-616.

[Faundez-Zanuy, 2004] Faundez-Zanuy M. (2004), On the vulnerability of biometric security systems. *IEEE Aerospace and Electronic Systems Magazine*, **19**, 6, pp. 3-8.

[Ferrer *et al.*, 2005] Ferrer M. A., Alonso J. B., Travieso C. M. (2005), Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 6, pp. 993-997.

[Fierrez-Aguilar *et al.*, 2005] Fierrez-Aguilar J., Garcia-Romero D., Ortega-Garcia J., Gonzalez-Rodriguez J. (2005), Adapted user-dependent multimodal biometric authentication exploiting general information. *Pattern Recognition Letters*, **26**, 16, pp. 2628-2639.

[Furnell *et al.*, 1995] Furnell S. M., Sanders P., Stockel C. T. (1995). The use of Keystroke Analysis for Continuous User Identity Verification and Supervision. *Proceedings of International Conference on Multimedia Communication*, pp. 189-193.

[Gaines *et al.*, 1980] Authentication by Keystroke Timing: Some Preliminary Results. dans. Rand Corporation.

[Gosset, 1908] Gosset W. S. (1908), The probable error of a mean. *Biometrika*, **6**, 1, pp. 1-25.

[Griess et Jain, 2000] On-line Signature Verification, Michigan State University

[Gupta et Cabe, 1997] A Review of Dynamic Handwritten Signature Verification, Computer Science Department James Cook University of North Queensland.

[Guven et Sogukpinar, 2003] Guven A., Sogukpinar I. (2003), Understanding users keystroke patterns for computer access security. *Computers and Security*, **22**, 8, pp. 695-706.

[Hastie *et al.*, 1992] A Model for Signature Verification, AT&T Bell Laboratories.

[He *et al.*, 2004] He C., Girolami M., Ross G. (2004), Employing optimized combinations of one-class classifiers for automated currency validation. *Pattern Recognition,* **37**, 6, pp. 1085-1096.

[Hocquet *et al.*, 2005] Hocquet S., Ramel J.-Y., Cardot H. (2005). Fusion of Methods for Keystroke Dynamic Authentication. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, pp. 224-229.

[IBG] International Biometric Group - www.biometricgroup.com/.

[Jain *et al.*, 2005] Jain A., Nandakumar K., Ross A. i. (2005), Score normalization in multimodal biometric systems. *pattern Recognition,* **38**, 12, pp. 2270-2285.

[Jain *et al.*, 2002] Jain A. K., Griess F. D., Connell S. D. (2002), On-line signature verification. *pattern Recognition,* **35**, 12, pp. 2963-2972.

[Jain *et al.*, 1997] Jain A. K., Lin Hong Pankanti S., Bolle R. (1997), An identity-authentication system using fingerprints. *Proceedings of the IEEE*, **85**, 9, pp. 1365-1388.

[Jain *et al.*, 2004] Jain A. K., Nandakumar K., Ross A. (2004), Score Normalization in Multimodal Biometric Systems. *pattern Recognition,* **38**, 12, pp. 2270-2285.

[Jain et Ross, 2002] Jain A. K., Ross A. (2002), Learning User-Specific Parameters In A Multibiometric System. *Proc. of International Conference on Image Processing (ICIP).*

[Jonathon Phillips *et al.*, 2000] An Introduction to Evaluating Biometric Systems. dans Computer, Vol. 33, pp. 56-63.

[Kacholia et Pandit, 2003] Kacholia V., Pandit S. (2003). Biometric Authentication using Random Distributions (BioART). *Canadian IT Security Symposium*.

[Kittler *et al.*, 1998] Kittler J., Hatef M., Duin R. P. W., Matas J. (1998), On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 3, pp. 226-239.

[Kohonen, 1989] Kohonen T. (1989). Self-Organization and Associative Memory. Springer-Verlag, 312 p.

[Kohonen, 1998] Kohonen T. (1998). Learning vector quantization *dans*: The handbook of brain theory and neural networks *Press M. ed.*, pp. 537-540.

[Kumar *et al.*, 2003] Kumar A., Wong C. M., Shen C., Jain A. K. (2003). Personal Verification Using Palmprint and Hand Geometry Biometric. *Audio-and Video-Based Biometric Person Authentication* pp. 668-678.

[Lai *et al.*, 2002] Lai C., Tax D. M. J., Duin R. P. W., Pekalska E., Paclik P. (2002). On combining one-class classifiers for image database retrieval. *3rd International workshop on multiple classifier systems*, Cagliari, Italy, pp. 212-221.

[Lee et Cho, 2005] Lee H.-j., Cho S. (2005), Retraining a Novelty Detector with Impostor Patterns for Keystroke Dynamics-Based Authentication. *Lecture Notes in Computer Science*, pp. 633-639.

[Leggett et Williams, 1988] Leggett J., Williams G. (1988), Verifying identity via keystroke characteristics. *International Journal of Man-Machine Studies*, **28**, 1, pp. 67-76.

[Leggett *et al.*, 1991] Leggett J., Williams G., Usnick M., Longnecker M. (1991), Dynamic identity verification via keystroke characteristics. *International Journal of Man-Machine Studies*, **35**, 6, pp. 859-870.

[Manevitz et Yousef, 2002] Manevitz L. M., Yousef M. (2002), One-class svms for document classification. *Journal of Machine Learning Research*, **2**, pp. 139-154.

[Mann et Whitney, 1947] Mann H. B., Whitney D. R. (1947), On a test of whether one of 2 random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, **18**, pp. 50-60.

[Markou et Singh, 2003] Markou M., Singh S. (2003), Novelty detection: a review--part 1: statistical approaches. *Signal Processing*, **83**, 12, pp. 2481-2497.

[Markou et Singh, 2003] Markou M., Singh S. (2003), Novelty detection: a review--part 2:: neural network based approaches. *Signal Processing*, **83**, 12, pp. 2499-2521.

[Martens et Dardenne, 1998] Martens H. A., Dardenne P. (1998), Validation and verification of regression in small data sets. *Chemometric and intelligent laboratory system*, **44**, pp. 99-121.

[Martens et Claesen, 1997] Martens R., Claesen L. J. M. (1997). Dynamic Programming Optimisation for On-line Signature Verification. *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pp. 653-656.

[Mason et Graham, 2002] Mason S. J., Graham N. E. (2002), Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, **128**, 484, pp. 2145-2166.

[Matsumoto *et al.*, 2002] Matsumoto T., Matsumoto H., Yamada K., Hoshino S. (2002). Impact of Artificial Gummy Fingers on Fingerprint Systems. *Proceedings of SPIE: Optical Security and Counterfeit Deterrence Techniques IV*, pp. 275-289.

[McLachlan et Peel, 2000] McLachlan G., Peel D. (2000). Finite Mixture Models. 456 p.

[McQueen, 1967] McQueen J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. *5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297.

[Mitchell, 1996] Mitchell M. (1996). An introduction to genetic algorithm. 224 p.

[Monrose et Rubin, 1997] Monrose F., Rubin A. (1997). Authentication via Keystroke Dynamics. *ACM Conference on Computer and Communications Security*, pp. 48-56
[Monrose et Rubin, 2000] Monrose F., Rubin A. D. (2000), Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems*, **16**, 4, pp. 351-359.

[Napier *et al.*, 1995] Napier R., Laverty W., Mahar D., Henderson R., Hiron M., Wagner M. (1995), Keyboard user verification: toward an accurate, efficient, and ecologically valid algorithm. *International Journal of Human-Computer Studies*, **43**, 2, pp. 213-222.

[NSTC, 2006] Measurement of Performance of Recognition Technologies. dans. National Science & Technology Council's Subcommittee on Biometrics.

[Obaidat et Sadoun, 1997] Obaidat S. M., Sadoun B. (1997), A Simulation Evaluation Study of Neural Network Techniques to Computer User Identification. *Inf. Sci.*, **102**, 1-4, pp. 239-258.

[Ord et Furnell, 2000] Ord T., Furnell S. M. (2000). User Authentication for Keypad-devices Using Keystroke Analysis. *Second International Network Conference*, Plymouth, UK, pp. 263-272.

[P.Pudil *et al.*, 1994] P.Pudil, Navovicova J., J.Kittler (1994), Floating search methods in feature selection. *Pattern Recognition Letters*, **15**, pp.

[Parzen, 1962] Parzen E. (1962), On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, **33**, pp. 1065-1076.

[Peacock *et al.*, 2004] Peacock A., Ke X., Wilkerson M. (2004), Typing Patterns: A Key to User Identification. *IEEE: Security & Privacy Magazine*, **02**, 5, pp. 40-47.

[Plamondon et Parizeau, 1988] Plamondon R., Parizeau M. (1988). Signature verification from position, velocity and acceleration signals : A comparative Study. *9th International Conference on Pattern Recognition (ICPR'88)*, pp. 260-265.

[Prabhakar et Jain, 2002] Prabhakar S., Jain A. K. (2002), Decision-Level Fusion in Fingerprint Verification. *pattern Recognition*, **35**, pp. 861-874.

[Rabiner, 1989] Rabiner L. R. (1989), A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, **77**, pp. 257-286.

[Richiardi et Drygajlo, 2003] Richiardi J., Drygajlo A. (2003). Gaussian Mixture Models for on-line signature verification. *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, pp. 115-122.

[Riedmiller et Braun, 1994] Riedmiller M., Braun H. (1994). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *the IEEE Internationale Conference on Neural Networks*, pp. 586-591.

[Schölkopf *et al.*, 2001] Schölkopf B., Platt J., Taylor S., Smola J., Williamson A. J. (2001), Estimating the support of a high-dimensional distribution. *Neural Computation*, **13**, 7, pp. 1443-1472.

[Schölkopf *et al.*, 2000] Schölkopf B., Williamson R. C., Smola A. J., Taylor J. S., Platt J. C. (2000), Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, **12**, pp. 582-588.

[Sheng *et al.*, 2005] Sheng Y., Phoha V. V., Rovnyak S. M. (2005), A parallel decision tree-based method for user authentication based on keystroke patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **35**, 4, pp. 826-833.

[Snelick *et al.*, 2003] Snelick R., Indovina M., Yen J., Mink A. (2003). Multimodal Biometrics: Issues in Design and Testing. *Fifth International Conference on Multimodal Interfaces*, pp. 68-72

[Stone, 1977] Stone M. (1977), An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion. *J. R. Stat. Soc.*, **38**, pp. 44-47.

[Stylianou *et al.*, 2005] Stylianou Y., Pantazis Y., Calderero F., Larroy P., Severin F., Schimke S., Bonal R., Matta F., Valsamakis. A. (2005). GMM-Based Multimodal Biometric Verification. *Proceedings of ENTERFACE'05 (Similar NoE Workshop on Multimodal Interfaces)*, pp.

[SVC, 2004] http://www.cs.ust.hk/svc2004/download.html. dans.

[Tax *et al.*, 1999] Tax D., Ypma A., Duin R. (1999). Support vector data description applied to machine vibration analysis. *Proceedings of the Fifth Annual Conference of the ASCI*, pp.

[Tax, 2001] Tax D. M. J. (2001). One-class classification, Delft University of Technology, http://ict.ewi.tudelft.nl/~{}davidt/thesis.pdf,

[Tax et Duin, 1999] Tax D. M. J., Duin R. P. W. (1999), Support vector domain description. *Pattern Recognition Letters*, **20**, 11-13, pp. 1191-1199.

[Tax et Duin, 2001] Tax D. M. J., Duin R. P. W. (2001). Combining one-class classifiers. *Proceedings of the second international workshop Multiple Combining Systems*, pp. 299-308.

[Tax et Duin, 2002] Tax D. M. J., Duin R. P. W. (2002), Uniform object generation for optimizing one-class classifiers. *J. Mach. Learn. Res.*, **2**, pp. 155--173.

[Tax et Duin, 2004] Tax D. M. J., Duin R. P. W. (2004), Support Vector Data Description. *Machine Learning*, **54**, 1, pp. 45-66.

[Thompson, 1979] Thompson R. M. a. K. (1979), Password Security: {A} Case History. *Communications of the ACM*, **22**, 11, pp. 594-597.

[Tisse, 2003] Tisse C. L. (2003). Contribution à la Vérification Biométrique de Personnes par Reconnaissance de l'Iris, Montpellier,

[Uludag *et al.*, 2004] Uludag U., Ross A., Jain A. (2004), Biometric Template Selection and Update: A Case Study in Fingerprints. *pattern Recognition*, **37**, 7, pp. 1533-1542.

[Vapnik, 1995] Vapnik V. N. (1995). The Nature of Statistical Learning Theory. Springer, 314 p.

[Wang *et al.*, 2003] Wang Y., Tan T., Jain A. K. (2003). Combining face and iris biometrics for identity verification. *AVBPA*, pp. 805--813.

[Wilcoxon, 1945] Wilcoxon F. (1945), Individual comparisons by ranking methods. *Biometrics Bulletin*, **1**, pp. 80-83.

[Wirotius, 2005] Wirotius M. (2005). Authentification par signature manuscrite sur support nomade Université François Rabelais de Tours, Tours,

[Wirotius *et al.*, 2004] Wirotius M., Ramel J.-Y., Vincent N. (2004). Improving DTW for Online Handwritten Signature Verification. *International Conference in Image Analysis and Recognition (ICIAR)*, pp. 786-793.

[Wirotius *et al.*, 2004] Wirotius M., Ramel J.-Y., Vincent N. (2004). Selection of Points for On-Line Signature Comparison. *International Workshop On Frontiers in Handwriting Recognition (IWFHR)*, pp. 503-508.

[Yam *et al.*, 2002] Yam C. Y., Nixon M. S., Carter J. N. (2002). On the Relationship of Human Walking and Running: Automatic Person Identification by Gait *ICPR*, pp. 1051-4651.1).

[Yan *et al.*, 2004] Yan J., Blackwell A., Anderson R., Grant A. (2004), Password Memorability and Security: Empirical Results. *IEEE Security and Privacy*, **2**, 5, pp. 25-31.

[Yan et Bowyer, 2005] Yan P., Bowyer K. (2005). Empirical Evaluation of Advanced Ear Biometrics. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.

[Yeung et Chow, 2002] Yeung D.-Y., Chow C. (2002). Parzen-Window Network Intrusion Detectors. *International Conference on Pattern Recognition*, pp. 385-388.

[Yu et Cho, 2004] Yu E., Cho S. (2004), Keystroke dynamics identity verification--its problems and practical solutions. *Computers and Security*, **23**, 5, pp. 428-440.

Principales publications

[Hocquet *et al.*, 2004] Hocquet S., Ramel J.-Y., Cardot H. (2004). Users authentication by a study of human computer interactions. *Huitième Forum de l'Ecole Doctorale : Ecole Doctorale " Santé, Sciences, Technologies "*, Tours.

[Hocquet *et al.*, 2005] Hocquet S., Ramel J.-Y., Cardot H. (2005). Fusion of Methods for Keystroke Dynamic Authentication. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, pp. 224-229.

[Hocquet *et al.*, 2005] Hocquet S., Ramel J.-Y., Cardot H. (2005). Utilisation de la dynamique de frappe pour l'authentification d'utilisateurs. *Premières Journées Informatique de la Région Centre (JIRC'2005)*, Blois.

[Hocquet *et al.*, 2006] Hocquet S., Ramel J.-Y., Cardot H. (2006). Estimation of User Specific Parameters in One-class Problems. *18th International Conference on Pattern Recognition*, Hong Kong, pp. 449-452.

[Hocquet *et al.*, 2006] Hocquet S., Ramel J.-Y., Cardot H. (2006). User Specific Parameters in One-class Problems: the Case of Keystroke Dynamics. *PRIS Pattern Recognition in Information Systems*, Paphos, Cyprus, pp. 127-135.

[Hocquet *et al.*, 2006] Hocquet S., Ramel J.-Y., Cardot H. (2006). Authentification par la dynamique de frappe. *15e congrès francophone de Reconnaissance des Formes et Intelligence Artificielle RFIA 2006.*

[Hocquet *et al.*, 2007] Hocquet S., Ramel J.-Y., Cardot H. (2007). BIOGINA: An authentication software based on Keystroke Dynamics. *European University Information System (EUNIS2007)* juin 26-27, 2007.

[Hocquet *et al.*, 2007] Hocquet S., Ramel J.-Y., Cardot H. (2007). User Classification for Keystroke Dynamics Authentication. *The Sixth International Conference on Biometrics (ICB2007)* May 23-24, 2007.

Authentification biométrique adaptative

Application à la dynamique de frappe et à la signature manuscrite

Résumé

L'objectif de cette thèse est d'étudier la mise en place, d'un système d'authentification biométrique, facile d'utilisation peu cher et performant.

Nous nous sommes donc intéressés à la biométrie comportementale qui permet de répondre à ces contraintes. Elle comporte néanmoins de nombreux inconvénients : la résolution de problèmes à une classe, l'évolution des caractéristiques des utilisateurs au cours du temps et une très grande variabilité entre les utilisateurs. La première partie de cette thèse passe en revue les principaux travaux réalisés dans le cadre de la classification à une classe et de la biométrie afin d'en dégager les verrous scientifiques qui restent à résoudre. La seconde partie présente nos propositions qui se basent sur l'utilisation d'une base de référence afin de faire en sorte que le système s'adapte au maximum et automatiquement à chaque utilisateur notamment par la détermination de paramètres personnalisés.

La dernière partie de ce manuscrit présente deux applications dans le domaine de l'analyse de la dynamique de frappe au clavier et de d'analyse de signatures manuscrites. La première application a été demandée par la société CAPMONETIQUE qui a financé ce travail. Cette partie, ainsi que les expérimentations effectuées sur l'authentification de signatures manuscrites démontrent l'intérêt de nos préconisations et valident la généricité puisque, dans les deux cas les performances des systèmes biométriques comportementaux augmentent de manière significative.

Mots-clés : biométrie, problème à une classe, dynamique de frappe, signature manuscrite, personnalisation du système

Abstract

The objective of this thesis is to study the creation of a biometric authentication system, easy to use, not expensive and with fair performances. We were interested, in a promising field of the biometry: the behavioral biometry.

This field has many disadvantages: the resolution of one-class problems, the evolution along the time of the user's characteristics and a great variability between users. The first part of this thesis reviews the state of the art of the one-class problem and his application to behavioral biometrics in order to identify the problem which still must be solved. The second part presents our proposition, which is based on the use of a reference database in order to adapt the system to each user by the determination of personalized parameters. The last part of this manuscript presents two applications: the analysis of keystroke dynamics and handwritten signatures. The first application was proposed by the company CAPMONETIQUE which financed this work. This part, as well as the experiments carried out on the authentication of handwritten signatures show the interest of our recommendations and validate the generality: in both cases the performances of the behavioral biometric systems increase significantly with our proposition.

Keywords: biometry, one-class problem, keystroke dynamics, on-line signature, system adaptation