

Chapitre 4

Méthodes de Fusion et Normalisation.

4.1 Introduction

Ajouter une modalité à un système biométrique, c'est ajouter une nouvelle source d'information. C'est pourquoi les systèmes multimodaux permettent d'obtenir de meilleurs résultats que les systèmes mono modaux correspondants. Ajouter des modalités permet également d'augmenter l'universalité du système. Car si l'utilisateur est réfractaire à une caractéristique biométrique donnée, l'information récoltée sur les autres modalités peut compenser la faible qualité de l'acquisition. La fusion biométrique est possible à chacun de ces niveaux : aux deux premiers niveaux, elle consiste souvent en une normalisation des données suivie d'une concaténation [289]; au niveau de la comparaison, elle se traduit par une combinaison des scores; enfin, au niveau de la décision, elle consiste en une confrontation des résultats de classification qui aboutit à une décision finale [290]. Nous traitons la question de la fusion et ses différents niveaux. Particulièrement la **fusion** des **caractéristiques** et des **scores** qui font objet de notre travail. Les principales méthodes de **normalisation des scores**, de fusion par **combinaison** et **classification** sont étudiées.

4.2 Les types de fusion

La fusion d'éléments biométriques peut se référer à de nombreux scénarios différents.

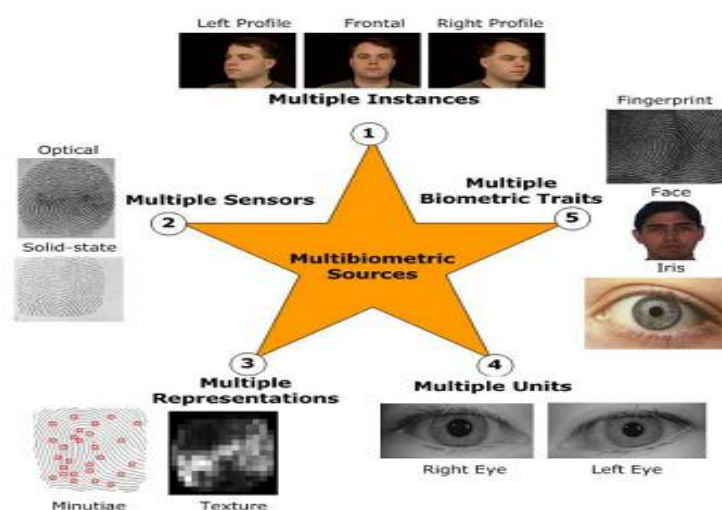


Figure 4.1 Sources de différents types de fusion de traits biométriques [3].

- **Systèmes multi algorithmes:** C'est le type de système le plus classique implicitement utilisé par de nombreuses approches. Les caractéristiques sont extraites via différents

algorithmes puis fusionnées. La fusion de caractéristiques extraites via un algorithme analysant les textures et un autre la forme d'un caractère biométrique entre dans ce cadre.

- **Systèmes multi échantillons:** Un capteur unique peut capturer plusieurs instances du même caractère biométrique dans le but de rendre plus robuste l'extraction des caractéristiques ou d'enrichir le modèle biométrique d'une personne. C'est le cas, par exemple, de plusieurs captures de visage d'une personne sous différents angles. L'utilisation de vidéos entre également dans ce cadre.
- **Systèmes multi capteurs:** Plusieurs capteurs permettent de capturer le même caractère biométrique sous différents angles. Ainsi la capture d'un visage à l'aide d'une caméra classique et d'une caméra infrarouge entre dans ce scénario. Ce type de système permet notamment la fusion au niveau capteur, ce que ne permettent pas d'autres systèmes comme les systèmes multi caractères.
- **Systèmes multi instances:** Ce type de système permet de capturer plusieurs instances du même caractère biométrique. L'acquisition de plusieurs empreintes digitales via le même capteur est l'exemple typique de ce type de système. Ces systèmes n'entraînent pas de surcoût de capteurs, ni le développement de nouveaux algorithmes. À ne pas confondre avec les systèmes multi échantillons.
- **Systèmes multi caractères:** Ce type de système combine différents traits biométriques d'un individu. Les fusions visage iris, ou visage empreinte digitale font partie de ce type d'approche. Ces systèmes nécessitent différents capteurs ainsi que des algorithmes dédiés à chaque caractère biométrique. Ce type de système a comme principale caractéristique que les caractères biométriques considérés peuvent être plus décorrélés que pour les systèmes multi capteurs.

La fusion de données issues de visages capturés via une caméra en lumière visible et une autre en lumière infrarouge entre dans le cadre des systèmes multi capteurs, où il est considéré que les deux captures sont issues de modalités différentes. Même si les deux captures sont sensiblement décorrélées (la chaleur émise par un visage n'est pas visible en lumière visible), la fermeture des yeux d'un individu est visible sur les deux modalités. À noter la présence de systèmes hybrides combinant plusieurs scénarios. Une revue de nombreux systèmes biométriques multimodaux développés peut être trouvée dans [291].

4.3 L'architecture des systèmes multimodaux.

Les systèmes multimodaux associent plusieurs systèmes biométriques et nécessitent donc l'acquisition et le traitement de plusieurs données. L'acquisition et le traitement peuvent se faire successivement, on parle alors **d'architecture en série**, ou simultanément, on parle alors **d'architecture en parallèle**.

L'architecture est en réalité surtout liée au traitement. En effet, l'acquisition des données biométriques est en général séquentielle pour des raisons pratiques. Il est difficile d'acquérir en même temps une empreinte digitale et une image d'iris dans de bonnes conditions. Il existe cependant certains cas où les acquisitions peuvent être faites simultanément lorsque les différentes données utilisent le même capteur par exemple les capteurs d'empreintes multi-doigts qui permettent d'acquérir plusieurs doigts simultanément ou même les empreintes palmaires. L'architecture est donc en général liée au traitement et en particulier à la décision. En effet la différence entre un système multimodal en série et un système multimodal en parallèle réside dans le fait d'obtenir un score de similarité à l'issue de chaque acquisition (fusion en série) ou de procéder à l'ensemble des acquisitions avant de prendre une décision (fusion en parallèle).

L'**architecture en parallèle (figure 4.3)** est la plus utilisée car elle permet d'utiliser toutes les informations disponibles et donc d'améliorer les performances du système. En revanche, l'acquisition et le traitement d'un grand nombre de données biométriques est coûteux en temps et en matériel, et réduit le confort d'utilisation. C'est pour cela que l'**architecture en série (figure 4.3)** peut être privilégiée dans certaines applications ; par exemple si la multimodalité est utilisée pour donner une alternative pour les personnes ne pouvant pas utiliser l'empreinte digitale. Pour la majorité des individus seule l'empreinte est acquise et traitée mais pour ceux qui ne peuvent pas être ainsi authentifiés on utilise un système à base d'iris alternativement.

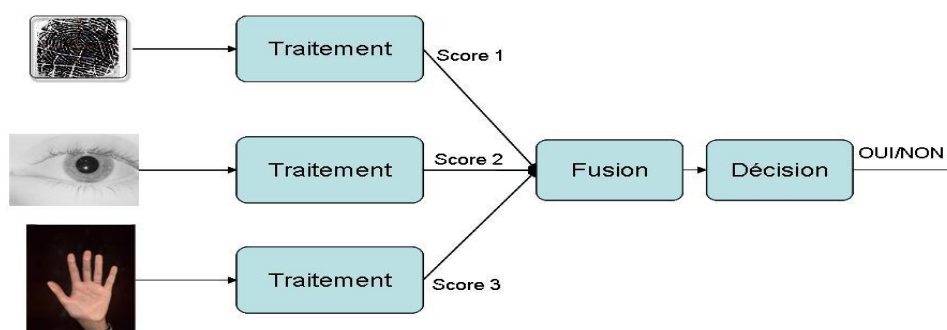


Figure 4.2 Architecture de fusion en parallèle

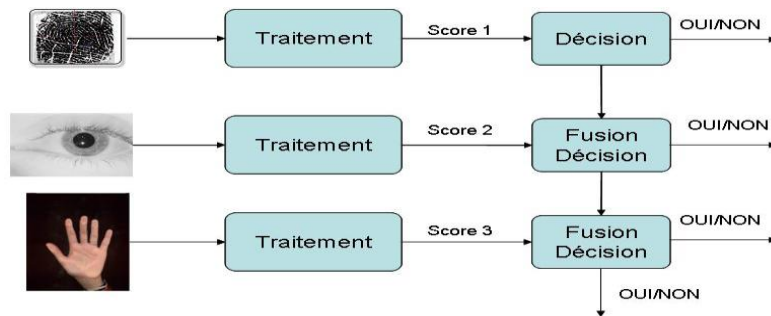


Figure 4.3 Architecture de fusion en série (incrémentale ou séquentielle)

4.4 Les niveaux de fusion [55]

La combinaison de plusieurs systèmes biométriques peut se faire à quatre niveaux différents : au niveau des **données**, au niveau des **caractéristiques** extraites, au niveau des **scores** issus du module de comparaison ou au niveau des **décisions** du module de décision (figure 4.4).

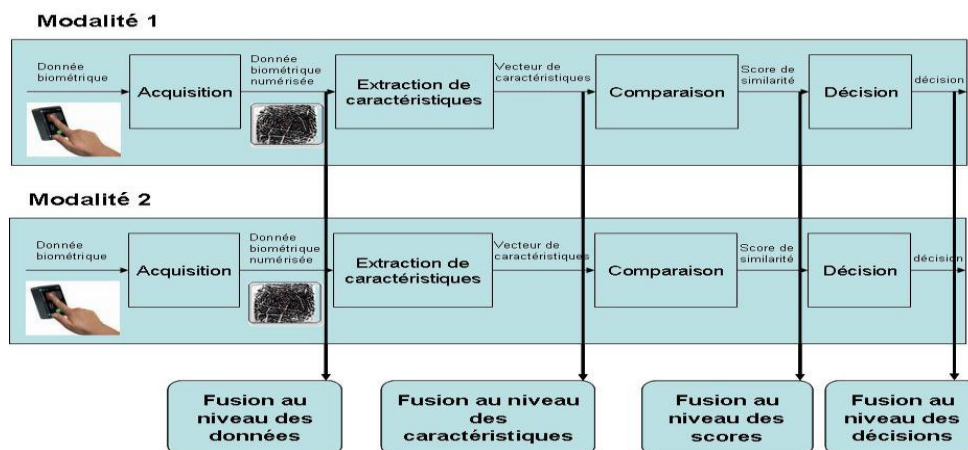


Figure 4.4 Les différents niveaux de fusion

Ces quatre niveaux de fusion peuvent être classés en deux sous-ensembles :

- la fusion pré-classification (avant comparaison),
- la fusion post-classification (après la comparaison).

4.4.1 La fusion pré-classification

La fusion pré-classification correspond à la fusion des informations issues de plusieurs données biométriques au niveau du capteur (images brutes) ou au niveau des caractéristiques extraites par le module d'extraction de caractéristiques.

1) Niveau du capteur (Sensor Level)

La fusion au niveau **capteur** est relativement peu utilisée car se faire uniquement si les diverses captures sont des instances du même trait biométrique obtenu à partir de **plusieurs capteurs** compatibles entre eux ou **plusieurs instances** du même trait biométrique obtenu à

partir d'un seul capteur. De plus, les captures doivent être compatibles entre elles et la correspondance entre les points dans les données brutes doit être connue par avance. Par exemple, les images de visage obtenues à partir de plusieurs caméras peuvent être combinées pour former un modèle **3D** du visage. Un autre exemple de fusion au niveau capteur consiste à mettre en mosaïque plusieurs images d'empreintes digitales afin de former une image d'empreinte digitale finale plus complexe. La fusion au niveau capteur n'est généralement pas possible si les instances des données sont incompatibles (par exemple, il est peut être difficile de fusionner des images de visages provenant de caméras ayant des résolutions différentes).

2) Niveau Caractéristiques (*Feature Level*)

La fusion au niveau des caractéristiques est moins limitée par la nature des données biométriques. Cependant une certaine homogénéité est nécessaire pour la plupart des méthodes de fusion au niveau des caractéristiques comme par exemple la moyenne de plusieurs "templates" d'empreintes ou de visage. Un exemple de fusion au niveau des caractéristiques qui ne nécessitent pas vraiment d'homogénéité est la concaténation de plusieurs vecteurs de caractéristiques avant le traitement par l'algorithme de comparaison. Par exemple, dans [292], **Jing et al** proposent une méthode de fusion de caractéristiques pour de la fusion de visage et d'empreinte palmaire. La fusion est effectuée par concaténation d'images obtenues par transformée de Gabor sur les images de visage et d'empreinte de la main. Mais la concaténation pose le problème de la dimension de l'espace de classification qui lorsqu'il augmente, rend plus difficile la tâche de classification.

Les méthodes de fusion pré-classification **sont assez peu utilisées** car elles posent un certain nombre de contraintes qui ne peuvent être remplies que dans certaines applications très spécifiques. En revanche, la fusion post-classification est très étudiée par les chercheurs.

4.4.2 La fusion post-classification

La fusion post-classification peut se faire au niveau des scores issus des modules de **comparaison** ou au niveau des **décisions**. Dans les deux cas, la fusion est en fait un problème bien connu de la littérature sous le nom de "Multiple Classifier systems".

1) Niveau Décision (*Decision Level*)

La fusion au niveau des décisions est souvent utilisée pour sa simplicité. En effet, chaque système fournit une décision binaire sous la forme **OUI** ou **NON** que l'on peut représenter par **0** et **1**, et le système de fusion de décisions consiste à prendre une décision finale en fonction de cette série de **0** et de **1**. Les méthodes les plus utilisées sont des méthodes

à base de **votes** telles que le **OR** (si un système a décidé 1 alors OUI), le **AND** (si tous les systèmes ont décidé 1 alors OUI) ou le vote à la **majorité** (si la majorité des systèmes ont décidé 1 alors OUI). On peut également utiliser des méthodes plus complexes qui pondèrent les décisions de chaque sous-système ou qui utilisent des classifieurs dans l'espace de décisions telles que **BKS** (Behaviour Knowledge Space). Dans [293], **Verlinde** présente un grand nombre de méthodes de fusion de décision. Ces méthodes de fusion au niveau des décisions sont très simples mais utilisent très peu d'information (0 ou 1).

2) Niveau Score (*Score Level*)

La fusion au niveau des scores est le type de fusion le plus utilisé car elle peut être appliquée à tous les types de systèmes (contrairement à la fusion pré-classification), dans un espace de dimension limité (un vecteur de scores dont la dimension est égale au nombre de sous-systèmes), avec des méthodes relativement simples et efficaces mais traitant plus d'information que la fusion de décisions. La fusion de scores consiste donc à la classification : **OUI** ou **NON** pour la décision finale, d'un vecteur de nombres réels dont la dimension est égale au nombre de sous-systèmes. Il existe un grand nombre de méthodes de fusion de scores qui seront présentée dans le paragraphe 4.3.5.

4.5 Les approches de fusion [55]

Nous allons maintenant nous intéresser aux méthodes de **fusion de scores**. Les méthodes de fusion de scores combinent les informations au niveau des scores issus des modules de comparaison comme indiqué sur la **figure 4.5**.

Un système de fusion est constitué de deux modules, un module de **fusion** et un module de **décision** (voir **figure 4.5**). Le problème devient donc un problème de classification à 2 classes (**OUI** ou **NON**, Client ou Imposteur) à partir d'un vecteur de nombre réels dont la dimension est égale au nombre de sous-systèmes du système **multi-algorithmes** .

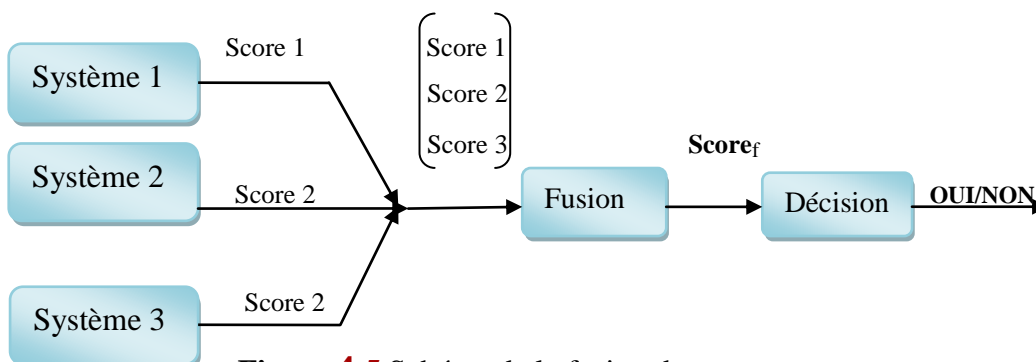


Figure 4.5 Schéma de la fusion de scores.

Il existe deux approches pour combiner les scores obtenus par différents systèmes.

La première approche est de traiter le sujet comme un problème de **combinaison**, tandis que l'autre approche est de voir cela comme un problème de **classification**. Il est important de noter que **Jain et al.** ont montrés que les approches par combinaison sont plus performantes que la plupart des méthodes de classification [294].

1) **Approche par combinaison de scores** : consiste à traiter le sujet comme un problème de **combinaison de scores** par des méthodes mathématiques de combinaison. Dans **l'approche par combinaison**, les scores individuels sont combinés de manière à former un unique score qui est ensuite utilisé pour prendre la décision finale. Afin de s'assurer que la combinaison de scores provenant de différents systèmes soit cohérente, les scores doivent d'abord être transformés dans un domaine commun : on parle alors de **normalisation de score**. [291]

On peut par exemple citer:

Kittler et al. [296] qui ont développés un cadre théorique pour combiner les informations d'identification obtenues à partir de plusieurs classifieurs en utilisant des méthodes simples de combinaison telles que la règle somme ("**sum rule**"), la règle "produit" ("**product rule**"), la règle maximum ("**max rule**"), la règle minimum ("**min rule**") et la règle médiane ("**median rule**"). Afin d'employer ces schémas, les scores de correspondance doivent être convertis en probabilités à postériori conformément à un utilisateur authentique ou imposteur. Ils considèrent le problème de classer un modèle d'entrée X dans l'une des m classes possibles (dans un système de vérification, $m = 2$) d'après l'information d'identification fournie par R classifieurs ou matchers différents.

Kwak et al [297] réalisent l'intégration floue de **Choquet** pour deux systèmes multi classifieurs. Le premier système combine des images traits du visage (yeux, nez et la bouche) et image globale de visage. Le deuxième système est basé sur des classifieurs opérant sur quatre ensembles de sous-image générée par décomposition en ondelettes.

Rasheed et al [298] utilisent l'intégrale floue de **Sugeno** pour la combinaison de scores pour un système multi classifieurs pour la décomposition d'un signal électromyographique (EMG).

Chia et al [299] utilisent une méthode hybride de calcul du minimum, maximum ou somme pour la combinaison des scores de deux systèmes d'authentification de visages et voix. **Yong**

Li et al [300] utilisent la somme pondérée pour la fusion par combinaison des scores. **Shukla et al** [301] proposent un calcul adaptatif et hybride utilisant la combinaison de scores par **logique floue** à base de l'intégrale de **Sugeno** ou de **Choquet**.

2) **L’approche par classification de scores** : consiste à voir cela comme un problème de *classification des scores*. Plusieurs classifieurs ont été utilisés pour classifier les scores de correspondance afin d’arriver à une décision. Dans **l’approche par classification**, un vecteur de caractéristiques est construit en utilisant les scores donnés en sortie par les systèmes individuels ; ce vecteur est ensuite attribué à une des deux classes : Client ou Imposteur. En général, le classifieur utilisé pour cette opération est capable d’apprendre la frontière de décision sans tenir compte de la manière dont le vecteur de caractéristiques a été généré. Ainsi, les scores en sortie de différents systèmes peuvent être non-homogènes (mesure de distance ou de similarité, différents intervalles de valeurs prises, etc...) et aucun traitement n’est requis avant de les envoyer dans le classifieur. On peut citer :

Verlinde et Chollet [302] combinent les scores provenant de deux modules de reconnaissance faciale et d’un module de reconnaissance de la parole avec l’aide de trois classifieurs : un premier classifieur utilisant la méthode des "k plus proches voisins" ("knearest neighbor algorithm", "**k-NN**") avec une quantification vectorielle, un deuxième classifieur basé sur **un arbre décisionnel** et un dernier classifieur basé sur un modèle de **régression logistique**.

Chatzis et al. [303] utilisent une méthode de regroupement ("clustering") appelée "**fuzzy k-means**" et une quantification vectorielle floue ("**fuzzy vector quantization**"), couplée à un classifieur de réseau de neurones de **RBF** médiane pour fusionner les scores obtenus à partir de systèmes biométriques basés sur des caractéristiques visuelles (visage) et acoustiques (voix).

Sanderson et al. [304] utilisent un classifieur basé sur une **Machine à Vecteurs de Support (SVM)** pour combiner les scores issus d’un module de reconnaissance faciale et d’un module de reconnaissance de la parole. Ils montrent que la performance d’un tel classifieur se détériore en la présence de conditions de bruit en entrée. Afin de surmonter ce problème, ils implémentent des classifieurs résistants au bruit structurel comme un classifieur linéaire défini par morceau ("**piece-wise linear classifier**") et un **classifieur Bayésien modifié**.

Wang et al. [305] qui considèrent que les scores provenant de modules de reconnaissance faciale et de reconnaissance par l’iris comme un vecteur de caractéristiques à deux dimensions. Une analyse discriminante linéaire de Fisher (**LDA**) et un classifieur par réseau de neurones combiné à une fonction de base radiale (**RBF**) sont ensuite utilisés pour la classification.

Ross et Jain [306] utilisent un **arbre décisionnel** et des **classifieurs discriminants linéaires** pour combiner les scores des modalités du visage, d'empreinte digitale et de géométrie de la main.

4.6 Normalisation de scores [55]

Les méthodes de normalisation de scores ont pour objectif de transformer individuellement chacun des scores issus des systèmes pour les rendre **homogènes** avant de les combiner. En effet, les scores issus de chaque système peuvent être de nature différente. Certains systèmes produisent des scores de **similarité** (plus le score est grand, plus la référence ressemble au test, donc l'utilisateur est un Client), d'autres produisent des **distances** (plus la distance est faible, plus la référence et le test sont proches, plus l'utilisateur est un Client). De plus chaque système peut avoir des intervalles de variations des scores différents, par exemple pour un système les scores varient entre 0 et 1 et pour un autre les scores varient entre 0 et 1000. On comprend bien la nécessité de normaliser les scores avant de les combiner. Les méthodes de normalisation présentées dans la suite, traitent des scores qui varient déjà tous dans le même sens (en général on considère tous les scores sous forme de similarité). Pour transformer des distances en similarité il existe deux solutions : l'inverse ou l'opposé. Dans toute la suite, nous considérerons que tous les scores à fusionner ont été transformés en scores de similarité (**scores Client > scores Imposteur**).

Les différentes techniques de normalisation de scores sont :

- Normalisation par la méthode **Min-Max** ;
- Normalisation par une fonction quadratique-linéaire-quadratique (**QLQ**) ;
- Normalisation par la méthode **Z-Score** ;
- Normalisation par la médiane et l'écart absolu médian (**MAD**) ;
- Normalisation par la méthode **tangente hyperbolique "Tanh"** ;
- Normalisation par une fonction **double sigmoïde**.

1 Normalisation par la méthode Min-Max

Cette méthode n'est pas robuste (insensible aux valeurs aberrantes). La normalisation **Min-Max** conserve la distribution de scores originale à un facteur d'échelle près et transforme tous les scores dans l'intervalle [0,1]. Le score normalisé Min-Max pour le score de test s_{ik} est donné par (4.1):

$$S'_{ik} = \frac{S_{ik} - \min(\{S_i\})}{\max(\{S_i\}) - \min(\{S_i\})} \quad (4.1)$$

s_{ik} le $k^{\text{ème}}$ score de sortie du $i^{\text{ème}}$ système, où $i = 1, 2, \dots, N$ et $k = 1, 2, \dots, M$

2 Normalisation par la méthode Z-Score

Cette méthode n'est pas robuste. De plus, la normalisation *Z-Score* ne garantit pas un intervalle commun pour les scores normalisés provenant de différents systèmes. Pour une distribution arbitraire, la moyenne et l'écart-type sont respectivement des estimateurs raisonnables de position et d'échelle, mais ne sont pas optimaux.

$$S'_{ik} = \frac{S_{ik} - \mu}{\sigma} \quad (4.2)$$

Où μ est la moyenne arithmétique et σ l'écart-type des données.

3 Normalisation par la médiane et l'écart absolu médian (MAD)

C'est une méthode robuste. Cependant, les estimateurs issus de la médiane et de la MAD ont une faible efficacité comparée aux estimateurs issus de la moyenne et de l'écart-type, c'est-à-dire que lorsque la distribution de score n'est pas gaussienne, la médiane et la MAD sont de pauvres estimateurs des paramètres de position et d'échelle. Ainsi, cette technique de normalisation ne conserve pas la distribution d'entrée et ne transforme pas les scores dans un intervalle commun.

$$S'_{ik} = \frac{S_{ik} - \text{median}}{\text{MAD}} \quad (4.3)$$

$$\text{MAD} = \text{median}(\{|s_i - \text{median}(\{s_i\})|\})$$

4 Normalisation par une fonction Quadratique-Linéaire-Quadratique (QLQ)

Snelick et al. [307] utilisent une fonction Quadratique-Linéaire-Quadratique (QLQ) pour normaliser des scores au préalable transformés dans l'intervalle $[0, 1]$ (s_{MM}) grâce à une normalisation *MinMax* (s_{MM}). Cette normalisation QLQ prend comme paramètres le centre c et la largeur w de la zone de recouvrement des distributions des scores Imposteurs et Client.

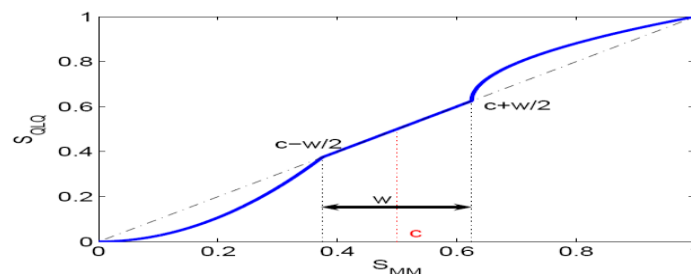


Figure 4.6 Normalisation QLQ.

La zone de recouvrement reste inchangée tandis que les autres régions sont transformées à l'aide de deux fonctions quadratiques par segment. Le score normalisé est donné par (4.4):

$$s_{QLQ} \begin{cases} \frac{1}{c-\frac{\omega}{2}} S_{MM}^2, & \text{si } s_{MM} \leq \left(c - \frac{\omega}{2}\right) \\ S_{MM}, & \text{si } \left(c - \frac{\omega}{2}\right) < s_{MM} \leq \left(c + \frac{\omega}{2}\right) \\ \left(c + \frac{\omega}{2}\right) + \sqrt{\left(1 - c - \frac{\omega}{2}\right) \left(s_{MM} - c - \frac{\omega}{2}\right)}, & \text{sinon} \end{cases} \quad (4.4)$$

5 Normalisation par la méthode tangente hyperbolique "Tanh"

Les scores normalisés sont donnés par (4.5):

$$s'_{ik} = \frac{1}{2} \left\{ \tanh\left(0.001 \frac{s_{ik} - \mu}{\sigma}\right) + 1 \right\}, \quad (4.5)$$

Où μ est la moyenne arithmétique, σ l'écart-type des données et \tanh la tangente hyperbolique. La méthode tangente hyperbolique met chaque score normalisé dans l'intervalle $[0, 1]$.

6 Normalisation par une fonction double sigmoïde

Cappelli et al. [308] ont utilisé une **fonction double sigmoïde** pour la normalisation de score dans un système biométrique multimodal qui combine différents systèmes d'empreinte digitale. Le score normalisé est donné par (4.6) :

$$s'_{ik} = \begin{cases} \frac{1}{1 + \exp\left(-2\left(\frac{s_{ik}-t}{r_1}\right)\right)} & \text{si } s_k < t, \\ \frac{1}{1 + \exp\left(-2\left(\frac{s_{ik}-t}{r_2}\right)\right)} & \text{sinon} \end{cases} \quad (4.6)$$

Où t est le point de fonctionnement de référence et r_1 et r_2 sont respectivement les bords gauche et droit, la **figure 4.7** montre un exemple d'une normalisation double sigmoïde, où les scores étant dans l'intervalle $[0, 300]$ sont transformés dans l'intervalle $[0, 1]$, avec $t = 200$, $r_1 = 20$ et $r_2 = 30$.

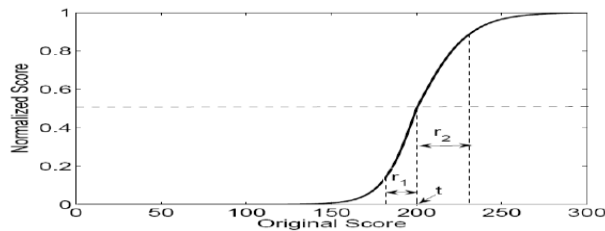


Figure 4.7 Normalisation double sigmoïde.

En général, t est choisi de façon à être une valeur appartenant à la région de recouvrement entre les distributions des scores client et imposteurs, et r_1 et r_2 sont fixés de telle sorte qu'ils correspondent respectivement aux extrémités de recouvrement entre les deux distributions vers la gauche et vers la droite de t (on peut également voir cela comme les valeurs correspondant respectivement au minimum des scores de similarité des clients et au

maximum des scores de similarité des imposteurs). Ce schéma de normalisation fournit donc une transformation linéaire des scores dans la région de recouvrement, tandis que les scores à l'extérieur de cette région sont transformés de manière non linéaire.

4.7 Approche par combinaison de scores

4.7.1 Méthode de combinaisons simples

Les méthodes de combinaisons de scores simples sont des méthodes très simples dont l'objectif est d'obtenir un score final s à partir des N scores disponibles s_i pour $i = 1$ à N issus de N systèmes.

Les méthodes les plus utilisées sont la moyenne, le produit, le minimum, le maximum ou la médiane.

- Combiner les scores par la moyenne consiste à calculer s tel que

$$s = \frac{1}{N} \sum_{i=1}^N s_i \quad (4.7)$$

- Combiner les scores par le produit consiste à calculer s tel que

$$s = \prod_{i=1}^N s_i \quad (4.8)$$

- Combiner les scores par le minimum consiste à calculer s tel que

$$s = \min (s_i) \quad (4.9)$$

- Combiner les scores par le maximum consiste à calculer s tel que

$$s = \max (s_i) \quad (4.10)$$

- Combiner les scores par la médiane consiste à calculer s tel que

$$s = \text{med} (s_i) \quad (4.11)$$

- La somme pondérée c'est une méthode un peu plus évoluée qui nécessite une adaptation par le réglage de paramètres.

$$s = \sum_{i=1}^N \omega_i s_i \quad (4.12)$$

La somme pondérée permet de donner des poids différents ω_i à chacun des systèmes en fonction de leur performance individuelle ou de leur intérêt dans le système multi-algorithmes.

4.7.2 Combinaison de scores par logique floue

La théorie de la **logique floue** (des sous-ensembles flous) a été introduite par Zadeh en 1965 [309] comme une extension de la logique binaire d'une part et une amélioration de la

logique multivaluée (admettant plusieurs valeurs de vérité) d'autre part. L'importance de la logique floue réside dans le fait qu'elle s'approche du **raisonnement humain** par l'intégration et le traitement du caractère approximatif, vague, imprécis ou flou de la connaissance humaine. Les termes linguistiques tels que **environ**, **moyenne**, **approximativement** sont de nature à donner un caractère flou aux phrases énoncées. Par exemple, la règle « si le prix est inférieur à 6000 DA, j'achète » sera intuitivement utilisable si le prix est de 6002 DA, mais elle ne pourrait être exploitée en logique classique puisque le prix indiqué ne satisferait pas la prémisse.

Dans [22] on réalise l'**intégration floue de Choquet** pour deux approches de fusion :

- La **première approche** est de combiner des images des traits du visage (yeux, nez et la bouche) et images globales de visage.
- La **deuxième approche** est à l'agrégation des classificateurs opérant sur quatre ensembles de sous-image générée par décomposition en ondelettes.

Mesure floue

Un jeu de fonction $g : P(Y) \rightarrow [0, 1]$ est appelée une **mesure floue** si les conditions suivantes sont remplies :

1. conditions aux limites: $g(\phi) = 0, g(Y) = 1$
2. monotonie : $g(A) \leq g(B)$, si $A \subset B$ et $A, B \in P(Y)$
3. continuité : $\lim_{i \rightarrow \infty} g(A_i) = g(\lim_{i \rightarrow \infty} A_i)$, si $\{A_i\}_i^\infty$ est une suite croissante d'ensembles mesurables.

A partir de cette définition, **Sugeno** [310] a introduit un soi-disant g_λ **mesure floue** qui est livré avec une propriété supplémentaire

$$g(A \dot{\cup} B) = g(A) + g(B) + \lambda g(A)g(B) \quad (4.13)$$

Pour tous les $A, B \subset Y$ et $A \cap B = \phi$, et pour certains $\lambda > -1$. Évidemment quand $\lambda=0$, le g_λ mesure floue devient une mesure de probabilité standard.

En général, la valeur de λ peut être déterminée en raison de l'état limite de la mesure floue g_λ . Cette condition pour $g(Y) = 1$. Par conséquent, la valeur de λ est déterminée par la résolution de ce qui suit:

$$g_\lambda(Y) = \frac{1}{\lambda} (\prod_{i=1}^n (1 + \lambda g^i) - 1) , \lambda \neq 0 \quad (4.14)$$

est l'équivalent de :

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda g^i) \quad (4.15)$$

Où $\lambda \in (-1, +\infty)$, $\lambda \neq 0$, et g^i est la valeur de **la fonction de densité floue**. La solution peut être facilement obtenue; évidemment on s'intéresse à la racine supérieure à -1.

2 Intégrale floue

L'intégrale floue de la fonction h calculée sur Y par rapport à une mesure floue g est définie sous la forme

$$\int_Y h(y) \circ g(\cdot) = \sup_{\alpha \in [0,1]} [\min[\alpha, g(\{y | h(y) \geq \alpha\})]] \quad (4.16)$$

a) Intégrale floue de Sugeno

Lorsque les valeurs des $h(\cdot)$ sont classés dans l'ordre décroissant,

$$h(y_1) \geq h(y_2) \geq \dots \geq h(y_n)$$

L'intégrale floue de **Sugeno** est calculée comme suit:

$$\int_Y h(y) \circ g(\cdot) = \max_{i=1:n} [\min(h(y_i), g(A_i))] \quad (4.17)$$

Où $A_i = \{y_1, y_2, \dots, y_i\}$ désigne un sous-ensemble d'éléments. Les valeurs de $g(A_i)$ pris en charge par la mesure floue sur les sous-ensembles correspondant d'éléments peut être déterminée de manière récursive sous la forme :

$$g(A_1) = g(y_1) = g^1 \quad (4.18)$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \quad (4.19)$$

Le calcul de la fonction de densité floue g^i sur la base des données sont assurées par la manière suivante

$$\begin{cases} g^i = \beta p_i, & i = 1 \\ g^i = (1 - \beta) p_i, & i = 2, 3, 4 \end{cases} \quad (4.20)$$

Où p_i est le taux de classification dans l'intervalle $[0, 1]$ pour chaque système.

$B \in [0, 1]$ est un facteur qui met en place un certain équilibre entre les résultats de la classification.

b) Intégrale floue de Choquet

Il a été démontré que (4.16) n'est pas une extension correcte de l'intégration de **Lebesgue** habituelle. En d'autres termes, lorsque la mesure est additive l'expression ci-dessus ne retourne pas l'intégrale au sens de **Lebesgue**. Afin de remédier à cet inconvénient, Murofushi et Sugeno [311] ont proposés un soi-disant intégrante floue de Choquet calculé de la manière suivante:

$$\int_Y h(y) d g(\cdot) = \sum_{i=1}^n [h(y_i) - h(y_{i+1})] g(A_i) h(y_{n+1}) = 0 \quad (4.21)$$

4.8 Approche par classification de scores

L'approche par classification de scores, cherche à prendre une décision à partir du vecteur de scores de dimension N . Le but des méthodes de fusion basées sur des classifieurs de chercher une séparatrice entre les 2 classes, **Client** et **Imposteur**, dans l'espace à N dimensions des scores. La séparatrice peut être plus ou moins complexe selon le classifieur utilisé. Toutes les méthodes classiques de reconnaissance des formes et d'apprentissage automatique peuvent être utilisées pour la fusion de scores qui est un problème de classification comme un autre. Dans le domaine des systèmes multi-algorithmes, dont l'objectif est d'utiliser un classifieur pour combiner plusieurs systèmes sources (ici ce sont des systèmes biométriques que l'on veut fusionner) afin de consolider la décision de classification, un grand nombre de méthodes ont été utilisées. Les méthodes utilisées dans notre travail sont :

- la méthode de **statistique de Fisher**,
- les machines à vecteurs de support (**SVM** : Support Vector Machine)
- les réseaux de neurones **RN**.

4.8.1 Fusion par méthode statistique de Fisher

La méthode **statistique** introduite ici, se base sur les travaux de **Fischer** [312] et fait usage d'une frontière de décision linéaire pour séparer deux populations données, à savoir les clients et les imposteurs dans notre cas. Dans le contexte particulier de l'authentification de personnes, **Duc et al** [289] utilisent eux aussi une règle de décision linéaire, mais l'optimisent en se fixant un critère dérivé de la théorie de **Bayes**. [313]

Envisageons à présent la règle de décision développée par **Fisher**. Elle se base sur le rapport de vraisemblance repris ci-dessous:

$$\frac{T(z|c)}{T(z|i)} > k \quad (4.22)$$

où k représente un seuil d'acceptation dont la valeur dépend du compromis **TFA/TFR** que l'on veut obtenir. Dans le problème qui nous préoccupe, $T(z|c)$ et $T(z|i)$ sont inconnus et doivent être estimés à partir des données d'apprentissage. Une hypothèse courante consiste à approcher les distributions réelles par des distributions normales à p variables $N_p(\mu_A, \Sigma)$, où $A = \{c, i\}$ représente la classe d'individus, μ_A le vecteur des scores moyens et Σ la matrice de covariance entre experts. En un premier temps, on supposera la matrice Σ indépendante de la classe d'individus. Sous de telles hypothèses, les fonctions de densité de probabilité s'écrivent sous la forme:

$$f_A(z) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (z - \mu_A)' \Sigma^{-1} (z - \mu_A) \right\} \quad (4.23)$$

Les paramètres μ_c , μ_i et Σ sont inconnus, mais peuvent être estimés à partir des données d'apprentissage, soit x les n_c données relatives aux accès clients et y , les n_i données relatives aux accès imposteurs (simulés). On a :

$$\hat{\mu}_c = \sum_{q=1}^{n_x} x_q / n_c \quad (4.24)$$

$$\hat{\mu}_i = \sum_{q=1}^{n_u} y_q / n_i \quad (4.25)$$

$$\hat{\Sigma}_c = \sum_{q=1}^{n_x} (x_q - \hat{\mu}_c)(x_q - \hat{\mu}_c)' / (n_c - 1) \quad (4.26)$$

$$\hat{\Sigma}_i = \sum_{q=1}^{n_y} (y_q - \hat{\mu}_i)(y_q - \hat{\mu}_i)' / (n_i - 1) \quad (4.27)$$

$$\hat{\Sigma} = [(n_c - 1)\hat{\Sigma}_c + (n_i - 1)\hat{\Sigma}_i] / (n_c + n_i - 2) \quad (4.28)$$

Notons que l'on tient compte ici, par l'intermédiaire de Σ , de $\hat{f}_c(z)/\hat{f}_i(z) \geq k$ qui peut exister entre experts. En combinant les équations (4.22) à (4.28), on peut réécrire sous la forme de $D_L(z) \geq \ln(k) = k^*$ où :

$$D_L(z) = (z - \frac{1}{2}(\hat{\mu}_c + \hat{\mu}_i))' \hat{\Sigma}^{-1} (\hat{\mu}_c - \hat{\mu}_i) \quad (4.29)$$

Fisher fut le premier à utiliser cette fonction à des fins de classification.

Comme $D_L(z)$ est linéaire en z , elle fut communément appelée **fonction linéaire discriminante** ou **LDF** (**L**inear **D**iscriminant **F**unction). Ainsi, la procédure à suivre pour vérifier l'identité d'un candidat, consiste à calculer $\hat{\mu}_c$, $\hat{\mu}_i$ et $\hat{\Sigma}$ à partir des données d'entraînement (ce qui est effectué une fois pour toutes), puis $D_L(z)$ et comparer celui-ci au seuil k^* donné. Si $D_L(z) \geq k^*$, le candidat est accepté comme client.

Dans le cas de distributions de scores clients et imposteurs ne satisfaisant pas l'hypothèse d'une covariance Σ unique, la règle de décision (4.29) peut se réécrire sous la forme $D_Q(z) \geq 2k^*$ où :

$$\begin{aligned} D_Q(z) = & (z - \hat{\mu}_i)' \hat{\Sigma}_i^{-1} (z - \hat{\mu}_i) \\ & - (z - \hat{\mu}_c)' \hat{\Sigma}_c^{-1} (z - \hat{\mu}_c) \\ & + \ln(|\hat{\Sigma}_i|/|\hat{\Sigma}_c|) \end{aligned} \quad (4.30)$$

$D_Q(z)$ est appelée **fonction quadratique discriminante** ou **QDF** (**Q**uadratic **D**iscriminant **F**unction). Cette règle permet d'obtenir de meilleurs résultats que ceux obtenus dans le cadre d'une fusion linéaire, mais requiert un ensemble d'apprentissage suffisamment étendu pour pouvoir évaluer Σ_c et Σ_i avec précision.

4.8.2 Fusion par méthode SVM

Les machines à vecteurs de support (Support Vector Machine : SVM), également appelés Séparateurs à Vaste Marge, introduits par Vapnik [288] est utilisée dans la fusion de données dans les références [16,314,315,316]. Ils ont pour but de définir un hyperplan de séparation entre les deux classes, hyperplan qui minimise l'erreur de classification sur un ensemble d'apprentissage. Cette méthode d'apprentissage supervisée peut apprendre une séparatrice plus ou moins complexe selon la nature du noyau choisi. Le noyau le plus simple est le noyau linéaire qui correspond à chercher un séparateur linéaire dans l'espace à N dimensions des scores. Le but des fonctions noyau est de transformer l'espace initial (des scores à N dimensions) en un espace de dimension **plus grande** dans lequel les données pourraient être linéairement séparables. La séparatrice est donc toujours linéaire dans l'espace transformé par la fonction noyau, mais ne l'est plus dans l'espace des scores.

Le but du SVM est donc de trouver une séparatrice qui minimise l'erreur de classification sur l'ensemble d'apprentissage mais qui sera également performante en généralisation sur des données non utilisées en apprentissage. Pour cela le concept utilisé est celui de marge (d'où le nom de séparateurs à vaste marge). La marge est la distance quadratique moyenne entre la séparatrice et les éléments d'apprentissage les plus proches de celle-ci appelés vecteurs de

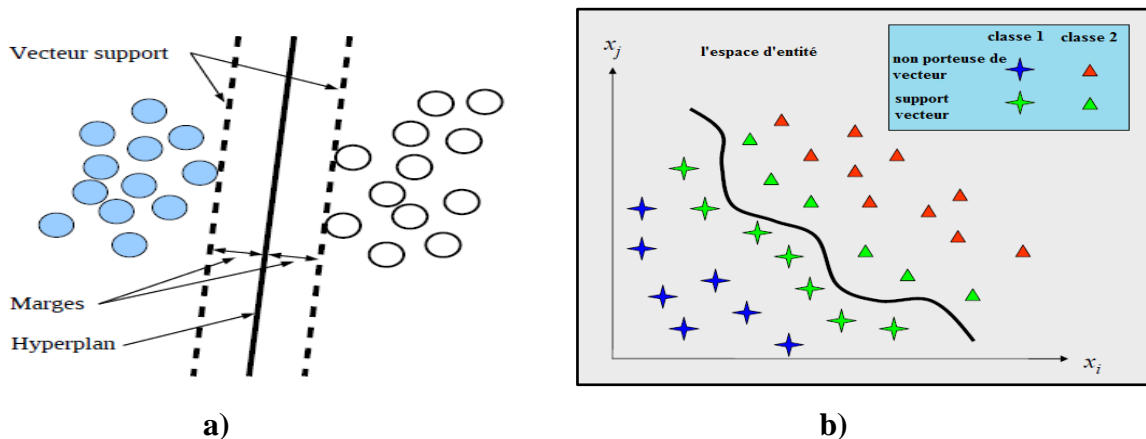


Figure 4.8 Séparation dans un espace à deux dimensions a) linéaire et b) non linéaire.

support (**figure 4.8**). Ces éléments sont appelés vecteurs de support car c'est uniquement sur ces éléments de l'ensemble d'apprentissage qu'est optimisée la séparatrice.

Le principe des SVM est présenté dans la section 3.3.2.

4.8.3 Fusion par Réseaux de Neurones Artificiels

Le principe général des **Réseaux de Neurones Artificiels (RNA)** est à l'origine inspiré de certaines fonctions de base des neurones naturels du cerveau.

Un réseau de neurones artificiel est généralement organisé en **plusieurs couches** :

- une couche d'entrée,
- une couche de sortie,
- des couches intermédiaires appelées couches cachées.

La présence de couches cachées permet de **discriminer** des classes d'objets non linéairement séparables. En général, un réseau de neurones est fondamentalement un classifieur. Il réalise un travail de classification pendant la phase d'apprentissage, et de classement lors de la reconnaissance. Mais on peut l'utiliser pour réaliser la fusion de données [317] pour séparer deux populations données, à savoir les clients et les imposteurs dans notre cas.

Les grands avantages des réseaux de neurones résident dans leur capacité **d'apprentissage automatique** (approximation universelle (**Cybenko, Hornik**)), ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes, tout en étant tolérant aux erreurs. Ils résident aussi dans leur capacité à prendre une décision à partir de critères **non formalisables explicitement**.

Structure du réseau perceptron multicouche MLP (Multi-Layer perceptron)

Les réseaux monocouches ont rapidement montré des limites car ils ne permettaient de traiter ni le cas des classes non linéairement séparables, ni le cas multi-classes. Le perceptron multicouche s'est imposé par la suite comme une réponse à ces problèmes, grâce à sa propriété d'approximateur universel [318], qui le rend capable d'approcher une fonction quelconque avec une précision arbitraire. Comme son nom l'indique, sa structure est organisée en couches (voir **figure 4.9**).

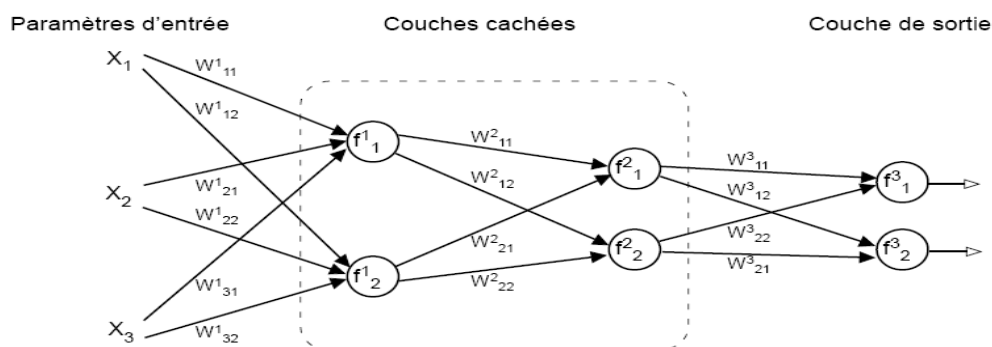


Figure 4.9 Exemple de perceptron multicouches

La **figure 4.9** illustre un exemple de perceptron multicouches avec 3 entrées, 2 couches cachées comportant 2 neurones chacune et une couche de sortie comportant 2 neurones.

Chaque neurone élémentaire est connecté à tous les neurones de la couche qui suit celle à laquelle il appartient. Les fonctions d'activation peuvent être quelconques, et le nombre de neurones en sortie dépend généralement du codage adopté. Par exemple, dans un problème de discrimination, chacun d'eux est généralement dédié à une classe donnée.

4.9 Conclusion

Ce chapitre nous a permis de connaître un certain nombre de méthodes de fusion de scores. Nous avons présenté les deux approches essentielles de fusion des scores la première est l'approche par combinaison. Cette dernière traite les scores séparément avant de les combiner et elle est basée sur des méthodes simples (les plus utilisées) telles que la moyenne, le produit, le minimum, le maximum ou la médiane ou bien par la logique floue.

Nous avons également montré qu'il y a différents niveaux de "complexité" pour les méthodes de combinaison.

La deuxième approche étudiée est basée sur la classification de scores qui cherche également à modéliser la séparatrice des classes dans l'espace multidimensionnel. Les méthodes utilisées dans notre travail sont :

- la méthode de **statistique de Fisher**,
- les machines à vecteurs de support (**SVM** : Support Vector Machine)
- les réseaux de neurones **RN**.

Une phase de normalisation préalable aux deux approches est réalisée c'est ce qui fait l'originalité de notre travail. Car dans les travaux antérieurs la normalisation se limitait à la première approche seulement (la fusion par combinaison). Cependant, il y a toujours de nombreuses questions ouvertes qui sont laissées sans réponse. Toutes ces méthodes ont des avantages et des inconvénients selon la complexité, le besoin de données d'apprentissage ou les paramètres à optimiser. Le choix d'une méthode va donc dépendre de l'application visée selon le compromis entre performance, simplicité, confort et quantité de données d'apprentissage disponible. Toutes ces méthodes ont été réalisées et comparées dans nos travaux sur la base de données **XM2VTS** [319].