CHAPITRE 2

MÉTHODOLOGIE EXPÉRIMENTALE

Ce chapitre a pour but de présenter la méthodologie expérimentale utilisée lors des expériences effectuées pour mesurer l'impact des divers facteurs des réseaux fuzzy ARTMAP. Le protocole expérimental utilisé traite des aspects relatifs aux simulations effectuées dans le cadre de ce projet de recherche.

Ce chapitre contient six sections. La première section présente les bases de données utilisées, soit les bases de données synthétiques et réelles. La deuxième section traite de l'emploi des diverses stratégies d'apprentissage utilisées lors des simulations. La section suivante présente les algorithmes de référence utilisés pour comparer les performances obtenues par les réseaux FAM. La quatrième section traite des techniques de normalisation utilisées lors du prétraitement des données pour les réseaux FAM. La cinquième section décrit les mesures de performance effectuées lors des simulations. Finalement, la dernière section présente le banc de test et les logiciels utilisés pour la réalisation des simulations.

2.1 Bases de données

Cette section décrit les bases de données synthétiques et réelles utilisées pour l'ensemble des simulations.

2.1.1 Bases de données synthétiques

Toutes les bases synthétiques utilisées sont formées de deux classes à 2 dimensions, équiprobables, dont la surface de répartition des classes est identique. Chaque base de données est composée de 300k patrons, distribués également entre deux classes. Ces bases de données sont divisées en trois bases de 100k patrons, soit la base

d'apprentissage, de validation et de test, chacune distribuée également entre les deux classes. La Figure 4 représente cette division.

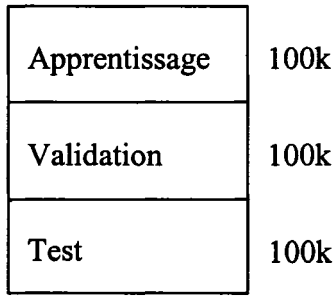


Figure 4 Séparation des bases de données synthétiques

Dans le but de présenter des résultats moyens face aux performances en généralisation du FAM, ces trois parties sont ensuite divisées en dix parties égales (BD_{APPi} BD_{VALIDi} BD_{TESTi} , *où* $i=[1,2,3,...,10]$) permettant dix réplications pour chaque problème. Chacune des 10 réplications est composée de trois bases de données de 10k patrons soit une base d'apprentissage, une base de validation et une base de test, chacune répartie également entre les deux classes.

Chaque réplication comporte 30 tests (S_i), chacun faisant augmenter graduellement la taille de la base d'apprentissage selon une progression logarithmique. Ainsi, le tout premier test (S_1) utilise les 10 premiers patrons (5 de chaque classe) de la première division de la base d'apprentissage (BD_{APP1}). Puis, le second test (S_2) utilise les 12 premiers patrons (6 de chaque classe) toujours de la première division de la base d'apprentissage (BD_{APP1}). Le tout jusqu'au $30^{ième}$ test qui utilise tous les patrons de la première division de la base d'apprentissage. Pour chacun de ces 30 tests, la base de test DB_{TEST1} est utilisée. Si une base de validation est nécessaire, DB_{VALID1} est utilisée. Finalement, le tout sera répété pour chacune des dix réplications.

Le Tableau II présente la taille des bases d'apprentissage utilisées lors des 30 tests effectués avec les bases de données synthétiques.

Tableau II

Taille des bases d'apprentissage pour les simulations avec les données synthétiques

TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage
#1	10	#11	108	#21	1172
#2	12	#12	136	#22	1486
#3	16	#13	174	#23	1886
#4	20	#14	220	#24	2395
#5	24	#15	280	#25	3038
#6	32	#16	356	#26	3856
#7	40	#17	452	#27	4892
#8	52	#18	572	#28	6210
#9	66	#19	726	#29	7880
#10	84	#20	922	#30	10000

Quatre types différents de bases de données synthétiques sont utilisés. Ces quatre types de bases de données se définissent comme suit :

BD_{μ} - Base de données respectant une distribution normale, avec une frontière de décision linéaire, dont le degré de chevauchement est dû au rapprochement des moyennes des classes. La variance de chaque classe reste fixe.

BD_{σ} - Base de données respectant une distribution normale, avec une frontière de décision linéaire, dont le degré de chevauchement est dû à l'augmentation des variances des classes. La moyenne de chaque classe reste fixe.

BD_{CiS} - Base de données provenant du problème *circle in square* [1]. Ce problème possède une erreur théorique nulle dont la frontière de décision est complexe.

BD_{P2} - Base de données provenant du problème P_2 [28]. Ce problème possède une erreur théorique nulle dont les frontières de décision sont complexes.

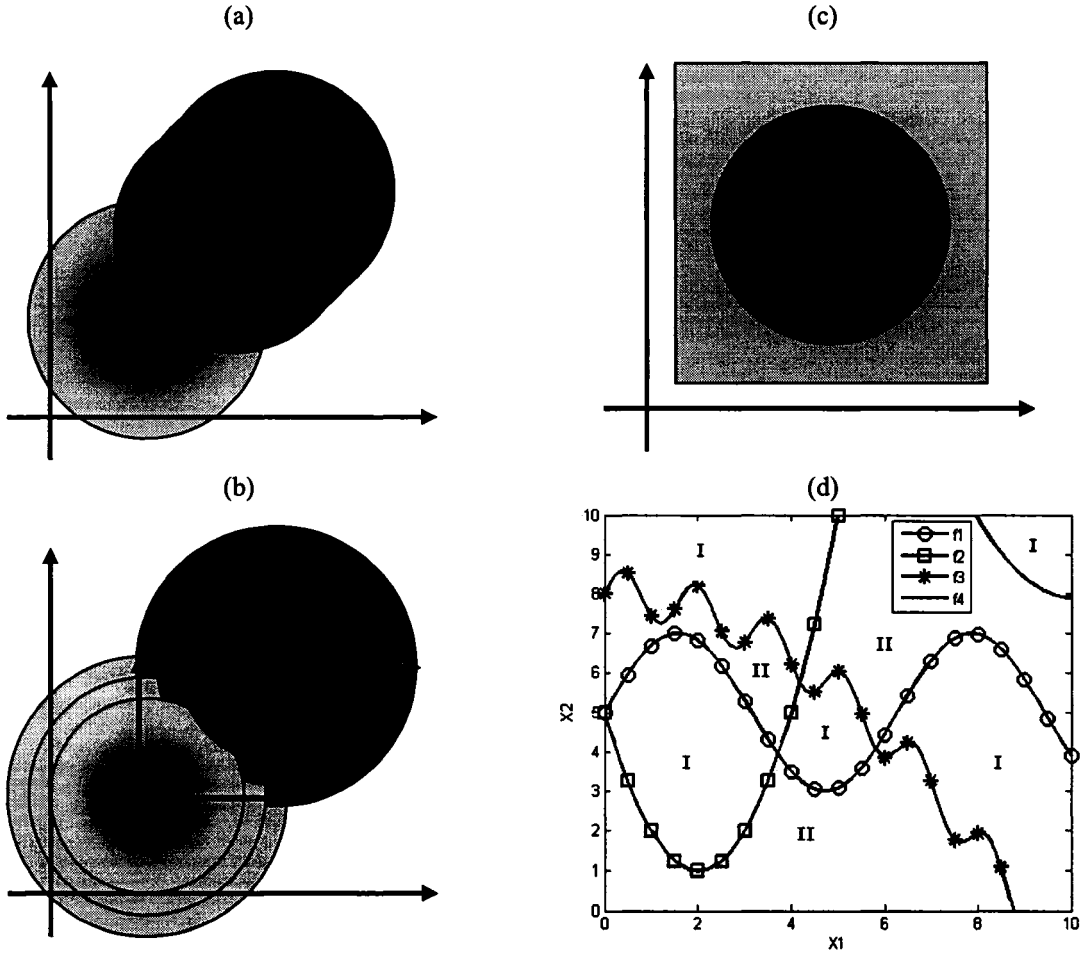


Figure 5 Représentation des bases de données synthétiques (a) DB_{μ}, (b) DB_{σ}, (c) DB_{CiS} et (d) DB_{P2}.

Les deux premiers types de bases de données représentent des problèmes dont le chevauchement est dû, respectivement, au rapprochement des moyennes des classes (DB_{μ}), et au rapprochement des variances des deux classes (DB_{σ}). Les deux derniers

types de bases de données (DB_{CIS} et BD_{P2}) représentent des problèmes de classification ne possédant aucun degré de chevauchement avec une ou des bornes de décision complexes. La Figure 5 présente ces quatre types de bases de données. Le degré de chevauchement des bases de données respectant une distribution normale (DB_μ et DB_σ) est graduellement augmenté de 1% à 25% d'erreur. Le Tableau III présente la valeur, au millième près, des paramètres utilisés pour la création des divers degrés de chevauchement lors du rapprochement des moyennes des classes (DB_μ). Le Tableau IV présente les paramètres utilisés pour la création des divers degrés de chevauchement lors de l'augmentation des variances des deux classes (DB_σ).

Tableau III

Paramètres des distributions normales pour les bases DB_μ

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\mu(\varepsilon = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 3\%)$	(0, 0)	(2.660, 2.660)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 5\%)$	(0, 0)	(2.326, 2.326)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 7\%)$	(0, 0)	(2.087, 2.087)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 9\%)$	(0, 0)	(1.896, 1.896)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 11\%)$	(0, 0)	(1.735, 1.735)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 13\%)$	(0, 0)	(1.593, 1.593)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 15\%)$	(0, 0)	(1.466, 1.466)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 17\%)$	(0, 0)	(1.349, 1.349)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 19\%)$	(0, 0)	(1.242, 1.242)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 21\%)$	(0, 0)	(1.141, 1.141)	(1, 1)	(1, 1)

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\mu(\varepsilon = 23\%)$	(0, 0)	(1.045, 1.045)	(1, 1)	(1, 1)
$DB_\mu(\varepsilon = 25\%)$	(0, 0)	(0.954, 0.954)	(1, 1)	(1, 1)

Tableau IV

Paramètres des distributions normales pour les bases DB_σ

Probabilité d'erreur	μ_1	μ_2	σ_1^2	σ_2^2
$DB_\sigma(\varepsilon = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)	(1, 1)
$DB_\sigma(\varepsilon = 3\%)$	(0, 0)	(3.290, 3.290)	(1.530, 1.530)	(1.530, 1.530)
$DB_\sigma(\varepsilon = 5\%)$	(0, 0)	(3.290, 3.290)	(2.000, 2.000)	(2.000, 2.000)
$DB_\sigma(\varepsilon = 7\%)$	(0, 0)	(3.290, 3.290)	(2.485, 2.485)	(2.485, 2.485)
$DB_\sigma(\varepsilon = 9\%)$	(0, 0)	(3.290, 3.290)	(3.011, 3.011)	(3.011, 3.011)
$DB_\sigma(\varepsilon = 11\%)$	(0, 0)	(3.290, 3.290)	(3.597, 3.597)	(3.597, 3.597)
$DB_\sigma(\varepsilon = 13\%)$	(0, 0)	(3.290, 3.290)	(4.266, 4.266)	(4.266, 4.266)
$DB_\sigma(\varepsilon = 15\%)$	(0, 0)	(3.290, 3.290)	(5.038, 5.038)	(5.038, 5.038)
$DB_\sigma(\varepsilon = 17\%)$	(0, 0)	(3.290, 3.290)	(5.944, 5.944)	(5.944, 5.944)
$DB_\sigma(\varepsilon = 19\%)$	(0, 0)	(3.290, 3.290)	(7.022, 7.022)	(7.022, 7.022)
$DB_\sigma(\varepsilon = 21\%)$	(0, 0)	(3.290, 3.290)	(8.322, 8.322)	(8.322, 8.322)
$DB_\sigma(\varepsilon = 23\%)$	(0, 0)	(3.290, 3.290)	(9.914, 9.914)	(9.914, 9.914)
$DB_\sigma(\varepsilon = 25\%)$	(0, 0)	(3.290, 3.290)	(11.90, 11.90)	(11.90, 11.90)

La Figure 6 présente les frontières de décision entre les classes de la base DB_{P2} originale [28]. Les fonctions mathématiques décrivant ces frontières sont présentées par les équations (2.1) à (2.4).

$$f_1(x) = 2 * \sin(x) + 5 \tag{2.1}$$

$$f_2(x) = (x-2)^2 + 1 \tag{2.2}$$

$$f_3(x) = -0.1 \cdot x^2 + 0.6\sin(4x) + 8 \tag{2.3}$$

$$f_4(x) = \frac{(x-10)^2}{2} + 7 \tag{2.4}$$

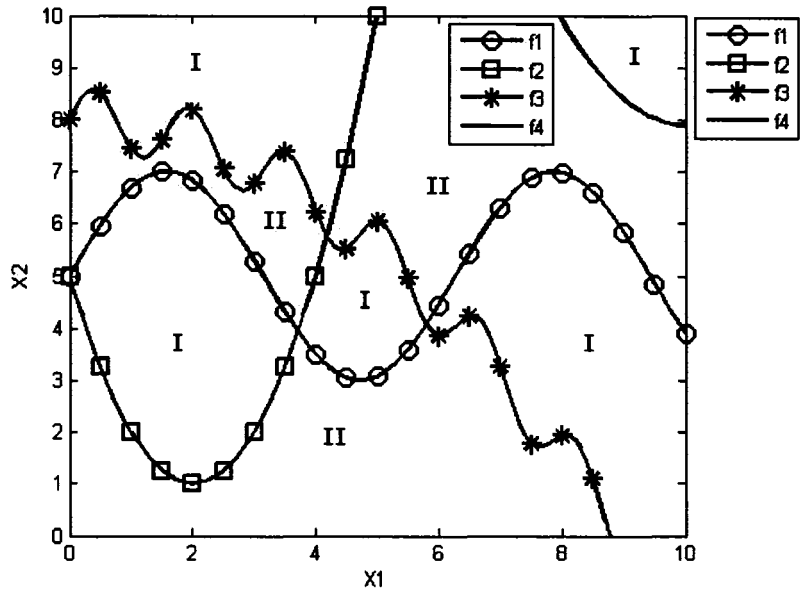


Figure 6 Frontière des classes de la base DB_{P2} [28]

Bien que la distribution de la base de données DB_{P2} soit uniforme et qu'il y ait le même nombre d'observations appartenant aux deux classes, la surface occupée par les observations des deux classes n'est pas identique. Le Tableau V présente le pourcentage de la surface occupée par chacune des classes de la base DB_{P2}.

Tableau V

Surface occupée par chaque classe de la base DB_{P2} originale

	Classe 1 *(I)*	Classe 2 *(II)*
Surface	52.2098%	47.7902 %

Pour obtenir la même probabilité *a priori* des deux classes, une légère modification de l'équation $f_4(x)$ (2.4) a permis d'obtenir une surface égale pour les observations des deux classes. La nouvelle fonction $f_4(x)$ est décrite par l'équation (2.5).

$$f_4(x) = \frac{(x-10)^2}{2} + 7.902 \tag{2.5}$$

2.1.2 Base de données réelles

La base de données réelles utilisée est la base NIST SD19. Elle est composée de 814255 images représentant des chiffres manuscrits (0 à 9). Cette base est divisée en huit sections *hsf*_{0,1,2,3,4,6,7,8}. La création de la base d'apprentissage et des bases de validation s'est effectuée en regroupant les bases de données *hsf*_{0,1,2,3}. Deux bases de test sont également utilisées : la base hsf_7 *(60 089 patrons)* étant la base de test standard de NIST SD19 et la base hsf_4 *(58 646 patrons)* étant une base bruitée, augmentant ainsi la difficulté de la classification.

La base de données NIST n'est pas une base de données équilibrée, ce qui veut dire que le nombre de patrons appartenant à une classe n'est pas égal à l'intérieur d'une série *hsf* de la base NIST SD19. Le Tableau VI présente le nombre de patrons contenus dans chacune des séries utilisées de la base NIST. Les séries *hsf*_{0,1,2,3} ont été

rééquilibrées car elles sont utilisées lors de la phase d'apprentissage alors que les séries *hsf*_{4,7}, utilisées pour la phase de test, ont été laissées telles quelles. Le nombre total d'images utilisées pour la base d'apprentissage et les trois bases de validation est de 195 000, soit 19 500 par classe. À partir de ces images, 132 caractéristiques sont extraites modélisant les aspects de chaque image. Ces caractéristiques ont été extraites par M. Oliveira lors de son Ph.D. à l'École de technologie supérieure de Montréal en collaboration avec le laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA) [29]. Elles représentent des ratios de concavité (78 caractéristiques), de contour (48 caractéristiques) et de surface (6 caractéristiques) [29]. Ainsi, dans les simulations avec la base NIST SD19, 313 735 patrons, composés de 132 caractéristiques, représentant 10 classes, sont utilisés. Le Tableau VII présente la répartition de ces patrons à travers les diverses bases de données utilisées.

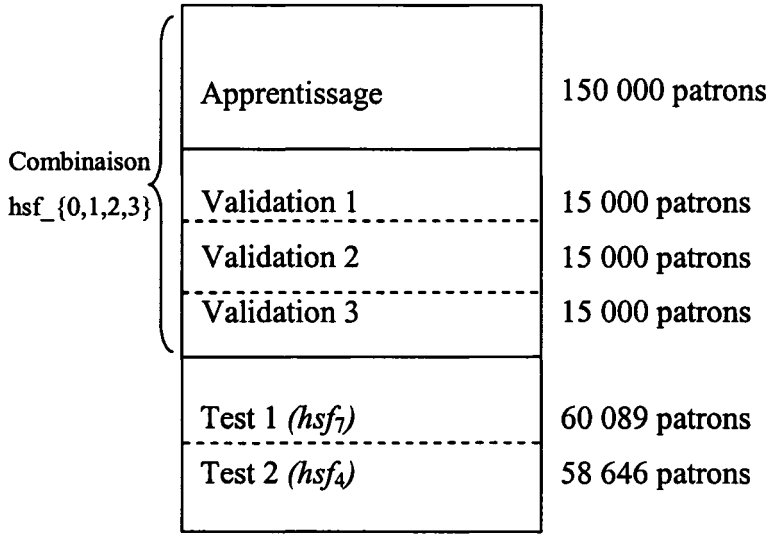
Tableau VI

Répartition des patrons de la base NIST SD19 dans les séries hsf

Class	*hsf*_{0,1,2,3}	hsf_4	hsf_7
0	22,971	5,560	5,893
1	24,772	6,655	6,567
2	22,131	5,888	5,967
3	23,172	5,819	6,036
4	21,549	5,722	5,873
5	19,545	5,539	5,684
6	22,128	5,858	5,900
7	23,208	6,097	6,254
8	22,029	5,695	5,889
9	21,619	5,813	6,026
Total	223,124	58,646	60,089

Tableau VII

Séparation des données de la base NIST SD19



Dans le but de présenter des résultats moyens face aux performances en généralisation du FAM, les simulations sont répétées 10 fois avec différents ordres de présentation définis aléatoirement. Chaque réplication comporte 15 tests provenant de la division de la base de données d'apprentissage en 15 tranches, où chaque tranche augmente graduellement la taille de la base d'apprentissage selon une règle logarithmique. Le premier test de la base NIST SD19 utilise une base de données d'apprentissage de 100 patrons (10 patrons par classe). Puis le deuxième test utilise une base de données d'apprentissage de 160 patrons (16 patrons par classe), et ainsi de suite jusqu'à ce que la base d'apprentissage soit de 150k patrons (15k par classe).

Le Tableau VIII présente les 15 grandeurs de la base d'apprentissage utilisées pour les simulations avec la base de données réelles.

Tableau VIII

Augmentation de la taille de la base d'apprentissage avec les données réelles

TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage	TEST No	Taille de la base d'apprentissage
#1	100	#6	1360	#11	18560
#2	160	#7	2290	#12	31290
#3	280	#8	3870	#13	52760
#4	470	#9	6520	#14	88960
#5	800	#10	11000	#15	150000

2.2 Stratégies d'apprentissage

Quatre stratégies d'apprentissage standard sont appliquées pour l'ensemble des bases de données. Ces quatre méthodes d'apprentissage sont :

a. Une époque (1EP);
b. Convergence des poids synaptiques ($CONV_W$);
c. Convergence des patrons d'apprentissage ($CONV_P$);
d. Validation hold-out (HV).

Le fonctionnement de ces méthodes est décrit à la section 1.2. De plus, nous avons développé une technique d'apprentissage spécifique pour les réseaux FAM. Cette technique utilise un algorithme d'optimisation afin d'améliorer les performances en généralisation du réseau FAM en sélectionnant de nouvelles valeurs pour les quatre paramètres internes du FAM. Cette technique utilise l'algorithme PSO pour la phase d'optimisation et une méthode d'apprentissage typique pour calculer la qualité. Ainsi, quatre techniques d'apprentissage spécialisées sont utilisées avec les quatre stratégies d'apprentissage standard et l'algorithme PSO. Ces quatre méthodes sont:

a. Optimisation PSO avec une époque "PSO(1EP)";
b. Optimisation PSO avec convergence des poids "PSO($CONV_W$)";
c. Optimisation PSO avec convergence des patrons d'apprentissage "PSO($CONV_P$)";
d. Optimisation PSO avec validation hold-out "PSO(HV)".

2.2.1 Stratégie d'apprentissage spécialisée avec optimisation par essaims particulaires

Lors des simulations avec les stratégies d'apprentissage spécialisées pour FAM utilisant PSO, l'algorithme PSO utilise 15 particules de recherche pour trouver la meilleure performance en généralisation sur l'ensemble des paramètres à optimiser. Les quatre paramètres internes des réseaux FAM sont optimisés soit: le choix (α), la vigilance ($\bar{\rho}$), le taux d'apprentissage (β) et le MatchTracking (ε). Ces paramètres sont optimisés pour chaque test, chaque base de données et chaque stratégie d'apprentissage. La plage d'optimisation utilisée pour chacun de ces paramètres est :

a. α : [0.00001, 1];
b. $\bar{\rho}$: [0, 1];
c. β : [0, 1];
d. ε : [-1, 1].

La vitesse maximale d'évolution d'une particule selon chaque paramètre est :

a. Vmax(α) : 0.1;
b. Vmax($\bar{\rho}$) : 0.1;
c. Vmax(β) : 0.1;
d. Vmax(ε) : 0.2.

L'évaluation de la performance de la première particule est effectuée à partir des valeurs par défaut utilisées avec les stratégies standard, soit: $\alpha = 0.01$, $\overline{\rho} = 0.0$, $\beta = 1.0$ et $\varepsilon = 0.001$. Toutes les autres particules sont initialisées aléatoirement sur la plage d'optimisation de chaque paramètre. Le nombre maximum d'itérations PSO est de 100 et quatre cycles d'optimisation sont effectués pour chaque expérience afin de minimiser l'impact de l'initialisation des particules. Il est à noter que l'algorithme PSO n'a jamais atteint la limite de 100 itérations lors de l'optimisation de tous les problèmes que nous avons testés.

Tel que décrit par l'algorithme 1 (voir section 1.3.1.1), au cours de l'optimisation PSO lorsqu'un réseau FAM obtient une meilleure valeur de qualité comparée à la meilleure qualité globale (*gbest)*, ce réseau devient le nouveau *gbest*. Lors des expériences d'optimisation avec les réseaux neuroniques FAM, nous avons remarqué que plusieurs réseaux différents obtiennent exactement les mêmes performances en généralisation. En utilisant cette connaissance à notre avantage, nous avons ajouté une nouvelle règle lors de l'établissement du meilleur réseau global :

> si un réseau obtient une valeur de qualité égale à celle de *gbest*, le réseau ayant la plus grande compression, soit le moins de catégories, est sélectionné comme étant l'optimum global (*gbest*).

Lors d'un cycle d'optimisation, lorsque que 10 itérations successives n'ont pas réussi à trouver un nouveau *gbest*, le cycle est arrêté. Le réseau FAM ayant obtenu la meilleure performance en généralisation sur la base de validation lors des 4 cycles est sélectionné comme étant le réseau optimal. Si plus d'un réseau a la même performance en généralisation, le réseau ayant le moins de catégories, soit le plus grand taux de compression, est sélectionné. Finalement, le meilleur réseau neuronique FAM trouvé lors des 4 cycles d'optimisation PSO est testé sur la base de test.

Les quatre stratégies d'apprentissage standard sont utilisées pour le calcul de la qualité à l'intérieur de l'algorithme PSO. Afin d'obtenir les meilleures performances possibles, l'ordre de présentation est également réparti au hasard entre chaque époque. Par contre, étant donné la grande demande en ressources pour l'optimisation des paramètres internes du réseau FAM, l'optimisation PSO est accomplie uniquement pour les bases de données $DB_{\mu}(1\%)$, $DB_{\mu}(9\%)$, $DB_{\mu}(25\%)$, $DB_{\sigma}(9\%)$, DB_{P2}, DB_{CIS} et uniquement avec la technique de normalisation MinMax.

Les stratégies d'apprentissage spécialisées pour FAM utilisant PSO sont également appliquées sur la base de données réelles NIST SD19. Étant donné la grande demande en temps de calcul de cet algorithme d'optimisation, une seule méthode d'apprentissage est utilisée sur cette base de données. Cette méthode est sélectionnée à partir des résultats obtenus par les tests effectués sur les données synthétiques.

2.3 Algorithmes de référence

Afin de bien étudier les résultats obtenus par les réseaux FAM, leurs performances sont comparées avec d'autres algorithmes de classification. Deux algorithmes différents sont utilisés comme référence, soit le classificateur quadratique Bayesien et la règle des *k* plus proches voisins. Malheureusement, ces deux classificateurs ne peuvent servir de référence dans certains types de problèmes. Le classificateur quadratique Bayesien est utilisé uniquement lors des simulations avec les bases DB_{μ} et DB_{σ} comme référence.

Il existe plusieurs avantages d'utiliser des données synthétiques, soit, entre autre, la connaissance parfaite de la distribution des classes. Grâce à cette connaissance il est possible de calculer l'erreur théorique ($\varepsilon_{théorique}$) pour chaque type de problème. Ainsi, les performances obtenues par les réseaux FAM et les performances des algorithmes de référence sont également comparées à l'erreur théorique de chaque type de base de données.

2.3.1 Classificateur quadratique Bayesien

Lors de l'utilisation de bases de données synthétiques respectant une distribution normale, le classificateur Bayesien est utilisé pour obtenir une performance de référence face aux résultats des simulations, et ce avec les bases DB_{μ} et DB_{σ}. Son utilisation comme algorithme de référence est basé sur le fait que ces bases de données *(DB_{μ} et DB_{σ})* respectent une distribution normale des données pour chaque classe.

Ce classificateur utilise une base de données d'apprentissage afin de calculer sa performance en généralisation. Cependant, l'erreur minimale théorique provenant de la connaissance des paramètres (moyenne et variance) des distributions normales n'est atteinte que si la base de données d'apprentissage tend vers l'infini. Quelques logiciels existent pour calculer l'erreur Bayesienne. Nous avons utilisé la plateforme PRTOOLS [31] de Robert P.W. Duin. L'annexe 1 présente le fonctionnement du classificateur Bayesien.

2.3.2 La règle du k plus proches voisins (kNN)

Lors des simulations avec *k*NN, le choix du nombre de voisins sur lequel s'effectue le vote pour déterminer la classification d'un patron de test est important. Plutôt que de fixer un nombre de voisins k pour toutes les simulations, le paramètre k est optimisé pour chaque simulation. Ainsi, chaque simulation vérifiera cinq différentes valeurs de k (1,3,5,7,9) sur la base validation, puis la valeur de k ayant obtenu la meilleure performance sera sélectionnée pour effectuer le test sur la base de test.

Le *k*NN est utilisé comme algorithme de référence pour toutes les bases de données synthétiques et réelles. L'annexe 2 présente le fonctionnement du classifieur *k*NN.

2.4 Normalisation des bases de données

Le réseau FAM requiert que toutes les caractéristiques des patrons présentés au réseau soient comprises dans l'intervalle [0, 1] inclusivement. Pour respecter cette spécificité du FAM, un prétraitement doit être effectué sur les bases de données. Ce prétraitement s'appelle la normalisation des données. Pour comprendre l'impact engendré suite à la méthode de normalisation utilisée, deux techniques de normalisation sont testées pour toutes les simulations effectuées, soit: la technique de normalisation MinMax et la technique de normalisation Centrée Réduite.

La normalisation MinMax est décrite par l'équation (2.6). Cette méthode de normalisation linéaire garantit que 100% des données normalisées sont comprises dans l'intervalle [0, 1]. Cependant, elle possède un désavantage. Si l'ensemble des données à normaliser comprend une donnée aberrante (donnée dont la fréquence d'occurrence est beaucoup moins élevée que toutes les autres) les données non aberrantes se retrouveront fortement compressées les unes sur les autres.

$$a'_{i,k} = \frac{a_{i,k} - \min_i}{\max_i - \min_i} \tag{2.6}$$

Où : $a'_{i,k}$ représente la valeur normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron

$a_{i,k}$ représente la valeur non normalisée de la $i^{ème}$ caractéristique du $k^{ième}$ patron

$\min_i$ représente la valeur minimale de la de la $i^{ème}$ caractéristique

$\max_i$ représente la valeur maximale de la de la $i^{ème}$ caractéristique

La normalisation Centrée Réduite (CRéduite) est décrite par l'équation (2.7). Cette méthode de normalisation linéaire garantit que 68% des données ($[\mu-\sigma, \mu+\sigma]$) seront normalisées dans l'intervalle [-1, 1]. Pour obtenir un taux de 99% des données normalisées dans cet intervalle, le dénominateur de l'équation (2.7) doit être remplacé

par $3\sigma_i$. Une fois les données normalisées par l'équation (2.7), il faut exécuter une translation des données de l'intervalle [-1, 1] à [0, 1].

Lors de l'utilisation de la méthode de normalisation Centrée Réduite, les données non comprises dans l'intervalle [-1, 1] sont mises à l'extremum (1 ou -1) le plus proche.

$$a'_{i,k} = \frac{a_{i,k} - \mu_i}{\sigma_i} \tag{2.7}$$

Où : μ_i représente la valeur moyenne de la $i^{ème}$ caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage

σ_i représente la variance de la $i^{ème}$ caractéristique de l'ensemble des patrons utilisés lors de la phase d'apprentissage

2.4.1 Bases de données synthétiques et réelles

Lors de l'utilisation des bases de données synthétiques, les deux techniques de normalisation sont applicables car les données ne sont pas définies dans l'intervalle [0, 1]. Avec la base de données réelles, toutes les caractéristiques sont déjà comprises dans l'intervalle [0, 1]. Les 132 caractéristiques extraites des images représentent des ratios de concavité, de contour et de surface [29].

Cependant, il est possible d'effectuer une normalisation des données, même si celles-ci sont déjà comprises dans l'intervalle [0, 1] et ce, afin d'améliorer la dispersion des données sur cet intervalle. En regardant les histogrammes de chaque caractéristique de la base NIST SD19, nous constatons que ces données pourraient bénéficier d'une normalisation, car elles sont majoritairement situées près de 0. La figure 8 présente l'histogramme de la $15^{ième}$ caractéristique extraite à partir de la base NIST SD19, lequel est représentatif de la majorité des 132 caractéristiques.

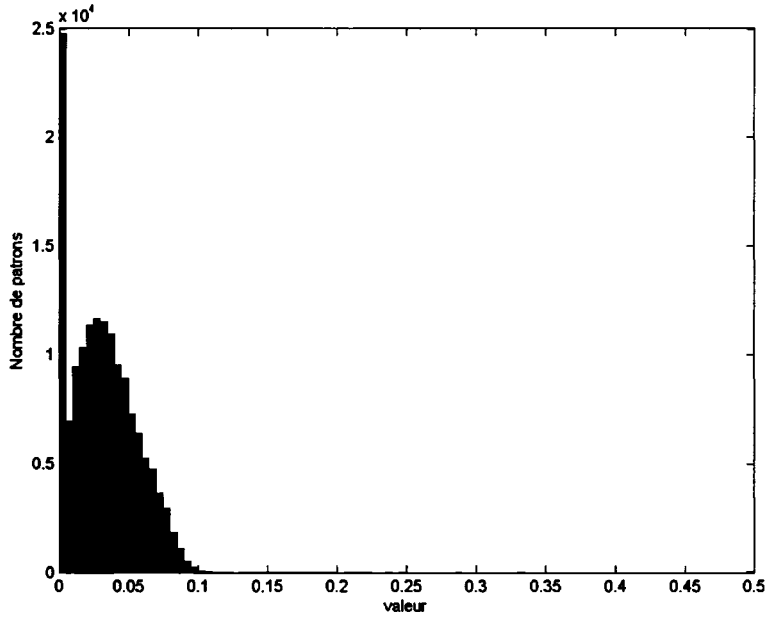


Figure 7 NIST SD19, caractéristique #15

Le chapitre 4 présente une discussion sur l'impact des deux techniques de normalisation appliquées sur les bases de données synthétiques. Le chapitre 5 contient une section portant sur les forces et les faiblesses des deux techniques de normalisation appliquées sur la base de données réelles NIST.

2.5 Mesures de performance

Pour chaque simulation, des mesures de performance sont effectuées. Ces mesures représentent la capacité de généralisation ainsi que les ressources utilisées par les réseaux FAM. La qualité du réseau FAM est obtenue par la mesure de l'erreur en généralisation et les ressources utilisées par le réseau FAM sont mesurées par le temps de convergence et le taux de compression.

Dans chaque cas, une mesure de dispersion est appliquée pour connaître la variabilité statistique des résultats. Pour mesurer la dispersion des résultats, l'équation de la déviation standard utilisée en mathématique statistique (2.8) est utilisée.

$$STD_{dev} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{2.8}$$

Où : n est la taille de la population

x_i est la valeur du ième individu en cours d'évaluation

$\overline{x}$ est la moyenne de la population

L'erreur en généralisation

L'erreur en généralisation est le rapport entre le nombre d'observations d'une base de test dont la classification obtenue est incorrecte sur le nombre total d'observations contenues dans cette base. Toutes les données de la base de test ne servent qu'une seule fois, soit pour le test final.

Le temps de convergence

Le temps de convergence montre le nombre d'époques d'entraînement qui ont été nécessaires au réseau FAM lors d'une simulation avant l'obtention de la condition d'arrêt. Une époque d'apprentissage signifie que toutes les observations de la base d'apprentissage ont été soumises au réseau.

Le taux de compression du réseau

Le taux de compression obtenu par le réseau FAM est calculé selon la formule (2.9).

$$C = \frac{\left|BD_{app}\right|}{Nb_{catégories}} \tag{2.9}$$

Où : $\left|BD_{app}\right|$ est la taille de la base d'apprentissage

$Nb_{catégories}$ est le nombre de catégories engendrées par le réseau

2.6 Banc de test

Le banc de test utilisé pour ces expérimentations a été créé et testé dans le cadre du laboratoire de recherche LIVIA de l'École de technologie supérieure de Montréal. Pour obtenir la convivialité du banc de test, le logiciel MatLAB est utilisé. Par contre, la performance de MatLAB au niveau de la rapidité de traitement laisse quelque peu à désirer. Pour obtenir un niveau de performance optimal, les calculs effectués par le réseau FAM sont codés en C. Ainsi, le banc de test se divise en deux grandes sections, soit la section MatLAB codée par M. Philippe Henniges et la section du code C codée par M. Dominique Rivard. La figure 1.7 présente le schéma des échanges de haut niveau du banc de test.

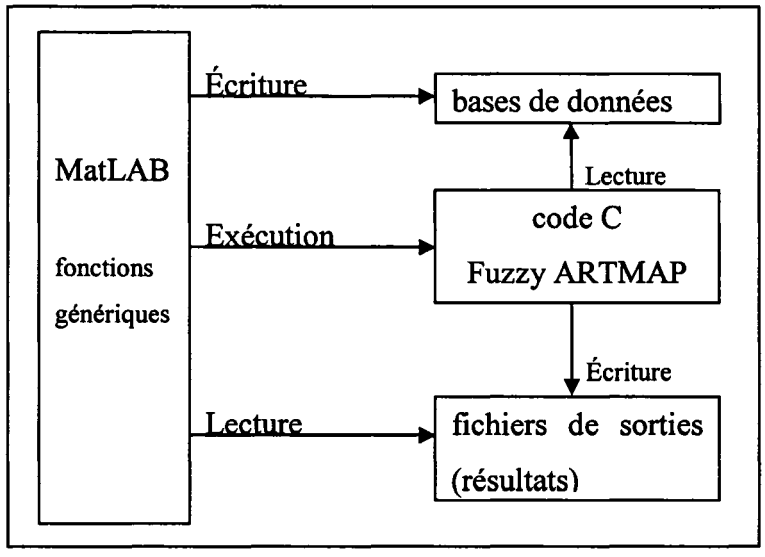


Figure 8 Schéma des échanges de haut niveau du banc de test

Bien que le noyau du code C du réseau FAM soit identique pour toutes les méthodes d'apprentissage, un exécutable a été compilé pour chaque méthode d'apprentissage. L'algorithme PSO a été codé sous MatLAB et il appelle la fonction C de son choix pour réaliser l'apprentissage d'un réseau FAM sur une base de données.

Lors des simulations PSO avec la base de données réelles NIST SD19, les simulations ont été exécutées sur une grappe d'ordinateurs (Beowulf cluster). L'algorithme PSO synchrone a été utilisé pour paralléliser le processus d'optimisation en utilisant 15 nœuds. Pour effectuer ces simulations, la création d'un nouveau code C utilisant les méthodes de traitement en parallèle MPI a été nécessaire.

De plus, la plateforme PRTOOLS [31] de Robert P.W. Duin a été utilisée pour la création des bases de données DB_{μ} et DB_{σ}, ainsi que pour les simulations avec les classificateurs *k*NN et quadratique Bayesien.