Université Paris-Nord Villetaneuse-Institut Galilée
Laboratoire d'informatique de Paris-Nord (L.I.P.N)

THÈSE

présentée pour obtenir

LE TITRE DE DOCTEUR D'UNIVERSITÉ

Spécialité Informatique

par

Fabrice BOSSAERT

Titre:

Approches Connexionnistes pour le Diagnostic des Systèmes Complexes: Application au Réseau Téléphonique

Soutenue le
devant le jury composé de

M.	Bechir	AYEB	Rapporteur
Mme	Sylvie	THIRIA	Rapporteur
M.	Younès	BENNANI	Directeur
M.	Philippe	DAGUE	Examinateur
Mlle	Elisabeth	DIDELET	Examinateur
M.	Daniel	STERN	Examinateur

Table des matières

Table des figures

Liste des tableaux

Titre : Approches connexionnistes pour le diagnostic des systèmes complexes: application au réseau téléphonique

Résumé:
Dans cette thèse, nous nous intéressons à l'utilisation et l'adaptation des techniques connexionnistes dans la réalisation des systèmes de diagnostic de systèmes complexes. Nous abordons le problème difficile de la sélection de variables et nous proposons deux nouvelles mesures de pertinence permettant de quantifier l'importance de chaque variable dans le système d'apprentissage. Ces techniques d'élagage permettent d'une part, d'ajuster la complexité du modèle à la difficulté du problème et d'autre part, de sélectionner un sous-ensemble de caractéristiques pertinentes. Nous abordons ensuite le problème du diagnostic des systèmes complexes en utilisant les techniques de sélection de variables comme pré-traitement des données. Nous proposons plusieurs systèmes connexionnistes multi-modulaires pour réaliser d'une part, des tâches de diagnostic sur le réseau téléphonique visant à assurer la détection d'incidents et d'autre part, l'identification de perturbations. Nous étudions ces deux problèmes du diagnostic (détection et identification) avec deux approches différentes : l'approche par modélisation connexionniste et l'approche par combinaison de modèles.

Title: Connectionist approaches for complex systems diagnosis: application to telephonic network

Abstract:
In this thesis, we are interested in the connectionist technique utilization and adaptation in the realization of diagnosis systems of complex systems. First, we deal with the difficult problem concerning variable selection, and we propose two new pertinence measures allowing variable importance quatification in the learning system. These pruning technics permit on the one hand, to fit the model complexity with on the order hand the problem diffculty, to select an underset of pertinent characteristics. Then we deal with the complex system diagnosis problem using variable selection technics as a data preprocessing. We propose various multi-modular connectionist systems to realize, on the one hand, diagnosis works on the telephonic network in order to assure incident detection and on the other hand, identification of perturbations. We deal with these two diagnosis problems (detection and identification) with two different approaches: the connectionist modelling approach, and the model combination approach.

Chapitre 1

Introduction

Il ne suffit pas de dire :
je me suis trompé;
il faut dire comment
on s'est trompé.

Claude Bernard 1813-1878

La gestion en temps réel du trafic téléphonique est nécessaire pour assurer une bonne qualité de service sur l'ensemble du réseau, en particulier en cas d'incidents. La complexité croissante du réseau et les demandes en terme de qualité et de service nécessitent l'introduction de traitements permettant d'automatiser cette gestion ou de construire des outils d'aide à l'opérateur. Cette thèse est centrée sur l'analyse et la mise en oeuvre de telles méthodes, et rentre dans le cadre d'un contrat de recherche avec le CNET et France Télécom sur le thème « Diagnostic de systèmes complexes, aide à la décision, détection de perturbation diagnostic et prévision ». En 1994, nous avons commencé à travailler en collaboration avec une équipe de l'Université Paris 6 sur l'utilisation de Réseaux de Neurones pour le diagnostic. Plus précisément, le but de cette thèse est le développement de réseaux de neurones pour réaliser d'une part des tâches de diagnostic sur le réseau visant à assurer la détection d'incidents et d'autres part des prévisions sur l'évolution à court terme d'indicateurs, afin d'anticiper l'état du réseau ou de détecter l'émergence de comportements caractéristiques de début d'incidents. Les deux méthodologies les plus courantes pour cela consistent à réaliser soit une discrimination à partir de données observées sur le sytème (des des indicateurs de trafic par exemple) soit à modéliser les comportements dynamiques du système et à mesurer lors de l'utilisation des écarts par rapport à un comportement nominal ou des similarités avec des comportements déjà modélisés. Dans le premier cas, il faudra définir des catégories ou classes correspondant aux situations que l'on veut détecter. L'apprentissage consistera à réaliser une discrimination des situations observées dans une de ces catégories. Dans le second cas il faudra identifier un ou plusieurs régime de fonctionnement afin de construire des modèles du système dans ces différents modes. Cette thèse a pour but de réaliser les objectifs suivants :

- La détection d'événements instantanés et la sélection de caractéristiques.
- La détection avec prise en compte de corrélations entre indicateurs.
- L'identification d'événements instantanés.

Cette thèse est divisée en 8 chapitres :

- Dans le chapitre 2, "Diagnostic et Reconnaissance des Formes":
  nous abordons dans un premier temps, une approche intuitive du diagnostic avec une décomposition du diagnostic en différentes tâches. Dans un deuxième temps, nous avons présenté le diagnostic dans le monde de l'intelligence artificielle, plus exactement le diagnostic à base de modèles, une des méthodologies la plus employée dans ce domaine. Pour finir, nous nous sommes intéressés au monde de la reconnaissance des formes, où différentes techniques sont présentées.
- Dans le chapitre 3, "Sélection et Extraction de caractéristiques":
  nous formalisons ce problème et présentons différentes techniques classiques

et connexionnistes parmis les plus repandues. Deux nouvelles mesures pour la sélection de variables: IIIE, et AINS, sont présentées. La première mesure est appliquée sur des modèles de type MLP et utilisée pour l'extraction de règles symbolique à partir de modèles connexionnistes. Notre seconde mesure a été développée afin de permettre la sélection de variables sur d'autres modèles connexionnistes que ceux de type MLP. Nous validons cette méthode, non seulement sur des MLPs, mais aussi sur des modèles de type RBF. Pour finir, nous terminons ce chapitre par la comparaison de nos deux mesures à différentes techniques connexionnistes.

- Dans le chapitre 4, "Extraction de règles":
  nous présentons un système d'extraction de règles issue de notre mesure IIIE et nous validons notre approche sur différents problèmes du domaine de l'apprentissage symbolique.

- Dans le chapitre 5, "Gestion en temps réel du trafic téléphonique":
  nous présentons dans un premier temps, le réseau téléphonique français. Nous introduisons ensuite les différents indicateurs de trafics qu'il nous a été possible d'utiliser, ainsi que le simulateur du trafic que nous avons eu à notre disposition. Enfin nous présentons de manière générale, les différentes méthodologies envisagées, pour répondre aux besoins de gestion.

- Dans le chapitre 6, "Génération d'alarmes dans uns réseau téléphonique":
  nous présentons une première phase de détection par un modèle univarié, où celle-ci se fait par intervalle de confiance. Dans un deuxième temps, nous proposons une détection à base de modèles multivariés et nous étendons cette détection en modélisant des situations perturbées. Cela nous a conduit à définir une mesure permettant d'augmenter le pouvoir discriminant des modèles par réduction de dimensions.

- Dans le chapitre 7, "Identification de perturbations":
  nous traitons ce problème, dans une première phase, à l'aide de combinaisons de modèles prédictifs. Après avoir présenté différentes techniques de combinaisons de modèles, nous avons étudié dans une deuxième phase, l'apport de la non-linéarité sur la linéarité.

- Dans le chapitre 8, "Conclusion et perspectives":
  nous présentons une synthèse de nos différents travaux, et définissons différents pôles de recherches que nous comptons développer.

- Dans le chapitre 9, "Annexes":
  nous présentons les différentes études effectuées sur les indicateurs du trafic téléphonique.

Chapitre 2

Diagnostic et Reconnaissance des Formes

Séronde : Il me semble que vous les placez autrement qu'ils le sont; que le coeur est du côté gauche et le foie du côté droit.

Sganarelle : Oui, cela était autrefois ainsi, mais nous avons changé tout cela.

[Poquelin1666]

Le Diagnostic est un problème très vaste. De nombreux pôles de recherche ont été développés pour le traiter, que soit dans le monde de l'Intelligence Artificielle ou bien celui de la Reconnaissance des formes. Dans ce chapitre, nous présentons dans un premier temps, une approche intuitive du diagnostic avec une décomposition du diagnostic en différentes tâches. Dans un deuxième temps, nous avons présenté le diagnostic dans le monde de l'intelligence artificielle, plus exactement le diagnostic à base de modèles, une des méthodologies la plus employée dans ce domaine. Pour finir, nous nous sommes intéressés au monde de la reconnaissance des formes, où les différentes techniques sont présentées.

2.1 Diagnostic

Pour définir le diagnostic, nous nous appuierons sur la définition donnée dans le Larousse médical [Domart et Bourneuf1981] en soulignant les points qui nous semblent importants. Diagnostic: "Temps de l'acte médical qui permet de **définir la nature** de la maladie **observée**. Le diagnostic est donc un temps capital puisqu'il permet de **classer** la maladie dans son cadre nosologique, d'en évaluer succinctement le pronostic vital ou fonctionnel et de choisir le traitement. Il est parfois très difficile, car il exige de la part du médecin une **analyse soigneuse** des éléments que recueille l'examen, groupant les **analogies** et faisant état des **dissemblances**; il exige donc un savoir qu'enrichit l'expérience, mais aussi un jugement sûr et parfois aussi une véritable intuition.
Il doit être distingué du terme diagnose, qui est l'art de définir les maladies par l'exposé concis mais suffisant de leurs symptômes caractéristiques et distinctifs.
Les temps successifs d'un diagnostic comportent : le diagnostic positif, le diagnostic différentiel, le diagnostic étiologique.
- Le diagnostic positif groupe les renseignements fournis par l'étude des faits commémoratifs immédiats et éloignés les indications données par l'examen clinique ... en un mot par un examen complet de tous les organes et appareils....
- Le groupement de tous ces éléments permettra de diagnostiquer la nature de la maladie et d'éliminer dans un diagnostic différentiel, les autres maladies présentant en partie des symptômes analogues.
- Enfin le diagnostic étiologique reconnaît la ou les causes de la maladie et permet parfois un traitement directement dirigé contre elles...."
Avant de traiter les points que nous considérons comme essentiels de cette définition, nous souhaitons ajouter un point qui pour nous est capital. En préambule à tout diagnostic, le patient détecte une anomalie et c'est donc lui qui commence le diagnostic, de plus le diagnostic porte sur un patient qui en fait correspond au système à diagnostiquer.
A partir de cette définition, nous pouvons donner de façon précise, ce qu' est pour nous le diagnostic.

- Système :
  Un système est un ensemble d'éléments interagissant entre eux et avec le monde extérieur dont l'état fluctue pendant le temps. Son rôle se traduisant par une tâche à accomplir, on peut prendre par exemple un homme, le système proprement dit ici est son corps au sens large et la tâche du corps est de garder en vie l'être humain. Un système a deux états possibles:

  l'état nominal: c'est le fonctionnement normal, le système répond correctement à la mission qu'il doit accomplir.

  l'état anormal, il regroupe les différents autres états que le système peut prendre, qui peuvent nuire à l'accomplissement de la tâche ou non, mais en

tout état de cause diffèrent des paramètres optimaux que l'on trouve pour un système en mode nominal.

On peut enfin définir le diagnostic proprement dit. Celui-ci se décompose en trois phases distinctes :

- Phase de détection
  Celle-ci doit répondre à la question: est-ce que le système est en mode nominal?

- Phase d'identification
  Cette phase consiste après avoir détecté une anomalie dans le fonctionnement du système, à déterminer dans quel état se trouve celui-ci. Pour un être humain, cette phase correspondrait à identifier la maladie dont souffre le patient.

- Phase opératoire
  Celle-ci doit trouver les actions à mettre en oeuvre, sachant l'état du système, pour restaurer l'état nominal du système.

Pour effectuer toutes ces phases, il faut a fortiori que l'on dispose de données sur le système, permettant de déterminer son état. Enfin avant d'élaborer toute action de diagnostic, il faut impérativement analyser ces données: en effet certaines données peuvent nuire au diagnostic et il est essentiel d'avoir une sélection de caractèristiques afin d'être dans les meilleures conditions possibles. Nous verrons de telles méthodes dans le chapitre 3.
En conclusion, pour établir un diagnostic, il est indispensable qu'il soit possible à partir de données d'identifier l'état du système pour établir des modèles de fonctionnements nominaux et anormaux.

2.2 Diagnostic et Intelligence Artificielle

Dans ce paragraphe nous ne traiterons pas toutes les méthodes de diagnostic, mais nous nous focaliserons sur une des méthodes la plus employée dans le monde du diagnostic, à savoir le diagnostic à base de modèles. Pour cela nous nous sommes basés sur [Dague et al. 1997]. Les connaissances utilisées pour ce type d'approche sont essentiellement basées sur la structure du système à diagnostiquer, à savoir les différentes relations entre les composants qui le constituent et leurs comportements proprement dit. Le comportement du système dépend des lois de la physique du domaine d'utilisation, que cela soit en termodynamique, en mécanique, etc....

Ces lois (plus exactement le comportement des composants du système) sont exprimées sous formes de contraintes. L'avantage majeur de ce type d'approche est que la connaissance nécessaire au diagnostic ne porte que sur le comportement normal du système à diagnostiquer: une fois le modèle de ce système établi, il suffit de comparer le système réel avec le modéle créé pour déterminer un mauvais fonctionnement de ce système.

Nous présentons la méthodologie employée, pour chaque étape du diagnostic décrit dans le paragraphe précédent.

- La détection:
  pour chaque composant ou ensemble de composants, on établit un modèle de son fonctionnement nominal, ce modèle permettant de prédire le comportant normal de ce(s) composant(s). La prédiction établie est comparée avec le comportement réel du composant (pour déterminer si celui-ci fonctionne correctement). Les valeurs prédites peuvent être qualitatives (signes, intervalles, etc...) ou encore numériques. Une fois l'anomalie détectée, il s'avère que l'information renseigne sur sa localisation: en effet, s'il apparaît une contradiction entre un modèle de comportement nominal et le modèle observé, c'est qu'il y a un ou plusieurs composants du modèle nominal en dysfonctionnement (un tel ensemble de composants, dont on est sûr que l'un au moins est en dysfonctionnement, est appelé un conflit).

- Localisation et/ou identification:
  pour déterminer le ou les composants rendant incohérent le modèle prédictif avec le modèle observé, il suffit de modifier les hypothèses du modèle prédictif jusqu'à suppression du ou des conflits; on obtient ainsi les différents composants en cause dans le dysfonctionnement. Pour cela, on utilise en pratique un solveur de problèmes couplé à un gestionnaire d'hypothèses (ATMS : "Assumption based Truth Maintenance System").

  Cette approche du diagnostic, qui s'appelle le diagnostic à base de cohérence, a été formulée la première fois par [Reiter1987]. On peut en avoir une description dans [Boubour1997]. Il apparaît souvent que plusieurs ensembles d'hypothèses (familles de diagnostic), peuvent être établis lors de la phase de localisation. Il faut alors, pour chaque famille de diagnostic ne conserver que le nombre d'hypothèses minimal. Pour cela on se sert d'algorithmes de génération d'ensembles d'hypothèses minimales: de tels algotithmes ont été proposés par [Reiter1987] et une version corrigée par [Greiner et al. 1989]. L'identification se fait par utilisation de modèles de faute: on cherche quels sont les modèles de faute des composants en dysfonctionnement qui sont cohérent avec les observations.

- Discrimination:
  elle consiste à déterminer les éventuelles mesures à effectuer, afin de minimi-

ser le nombre de diagnostics, ou plus exactement déterminer l'information qui permettrait de discriminer au mieux les diagnostics restants. Il existe des techniques basées sur des probabilités.

- Cycle de diagnostic:
  il s'agit de réitérer les phases précédentes. De cette façon, on détermine de nouveaux conflits qui permettent de supprimer certain diagnostics, ne reflétant pas ces nouveaux conflits. On peut dans cette phase faire un classement de probabilité sur les diagnostics. Pour cela il s'agit d'ajouter des modèles de comportement incorrect munis de probabilités a priori. En ce qui concerne le moment où il faut arrêter ce cycle, il reste à déterminer un compromis entre fiabilité du diagnostic, et temps de calcul pour affiner le diagnostic.

En conclusion:
Nous avons grossièrement décrit le principe du diagnostic à base de modèle, mais nous n'avons pas cité toutes les méthodes utilisées pour gérer ces phases. Cependant nous pouvons exhiber une difficulté inhérente à ce type d'approche: en fait la conception du modèle reste le problème majeur pour ce type de technique et il existe peu de moyen d'automatisation [Dague1994].

2.3 Diagnostic et Reconnaissance des Formes

Les algorithmes de la reconnaissance des formes, ont comme données initiales un ensemble d'individus et un ensemble de classes. Chaque individu est associé à une classe et le principe est alors le suivant: on regroupe les individus d'une même classe et on détermine un prototype de cette classe et ceci pour toutes les classes. L'ensemble des individus étiqueté sera appelé la base d'apprentissage.
La manière d'identifier un prototype dépend non seulement de la classe à prototyper, mais du souci d'essayer d'avoir des prototypes différents pour chaque classe. Il faut de plus discriminer des frontières entre les classes afin de ne pas avoir de confusions. Pour pouvoir faire du diagnostic à l'aide des techniques de la reconnaissance des formes, un protocole doit être suivi :

- Étape 1 : Déterminer les classes à traiter.
  Pour cela, il faut identifier les différents états dans lesquel peut se trouver le système. Ces différents états correspondront aux classes.

- Étape 2 : Récolter des individus.
  Dans cette étape, l'idée est d'observer le système dans les différents états possibles et à des moments différents afin d'obtenir un ensemble d'individus pour chaque classe.

- Étape 3 : Création des prototypes.
  Ici, on met en oeuvre l'algorithme proprement dit de reconnaissance des formes et suivant l'algorithme choisi, on obtiendra différents prototypes.

- Étape 4 : Utilisation.
  En observant le système, on obtient un individu que l'on compare à chaque prototype et le prototype le moins éloigné en terme de distance, fera correspondre sa classe à l'individu.

Pour formaliser, ces étapes, on peut définir un individu x comme un vecteur ayant n composantes, celles-ci étant les valeurs des différents capteurs du système étudié. On dispose de M classes $\{w_1, w_2, \cdots, w_M\}$ correspondant aux différents états du système. L'algorithme détermine k prototypes de chaque classe
$\{p^1_{w_i}, p^2_{w_i}, \cdots, p^k_{w_i}\}$.
Ces prototypes sont aussi des vecteurs très souvent de dimension plus faible que les individus x.
Il est très courant d'effectuer une sélection de variables permettant de déterminer un sous-espace de dimension inférieure à n, afin d'obtenir les meilleurs prototypes possibles. Dans ces cas là, on ne calcule pas la distance entre l'individu x et un prototype $p^k_{w_i}$, mais on projette l'individu dans l'espace du prototype, et on obtient l'individu x^P. L'application de cette technique à la tâche de diagnostic est triviale. En effet, pour ce qui est de la phase de détection, il suffit de déterminer les deux classes normales et anormales et pour l'identification, représenter par plusieurs classes les différents états perturbés du système. En ce qui concerne la phase opératoire, on utilise le plus souvent des techniques classiques basées sur des systèmes de règles de décision, par exemple les systèmes experts. Nous allons maintenant décrire succinctement différentes méthodes de reconnaissance de formes. Nous conseillons au lecteur l'ouvrage [Dubuisson1990], où sont présentées ces techniques de manière plus approfondie et d'autres techniques non citées ici.

2.3.1 Technique des k plus proches voisins: KNN (K-Nearest Neighbours)

2.3.1.1 Règle du plus proche voisin: 1PPV (1NN)

Soit X l'ensemble d'apprentissage composé de n vecteurs indépendants $x_1, \cdots, x_t, \cdots, x_n$ (x_t représentant l'état du système à l'instant t) et une distance $d(.,.)$ donnée sur l'espace des formes. Les classes des élements de X sont connues, nous désignerons la classe de l'élément x_t par $w(x_t)$. Soit x une forme à identifier, la technique du 1PPV: plus proche voisins se formule de la façon suivante:

$$w^*(x) = w(x_{1PPV}) \text{ si } d(x, x_{1PPV}) = \underset{j=1,n}{Min}(d(x, x_j))$$

où $w^*(x)$ est la classe d'affectation estimée de x et x_{1PPV} est le plus proche voisin de x.

2.3.1.2 Règle des k plus proches voisins: KPPV (KNN)

Dans ce cas la technique consiste à affecter l'object inconnu x à la classe la mieux representée parmi les k voisins les plus proches de x. On peut ainsi, en faisant varier k, optimiser la règle de décision.

2.3.2 Techniques de Groupement: "Culstering"

Les techniques de "Clustering" ont pour but, de réduire le nombre d{elements de l'ensemble d'apprentissage, d'où une diminution du nombre de distances à calculer. Nous présentons par la suite les deux techniques les plus utilisées dans le domaine de la reconnaissance des formes : k-moyennes (k-means) [Macqueen1967] et LBG [Linde et al. 1980].

2.3.2.1 Méthode des k-moyennes: k-means[Macqueen1967]

K-moyennes ou "k-means" est une méthode qui consiste à effectuer une suite de regroupements en agrégeant à chaque étape les formes ou les groupes de formes les plus proches. L'algorithme général est le suivant:

1- Initialisation
- . choisir au hasard k prototypes $\{p^1, \cdots, p^k\}$
- . on pose $m = 0$
- . on fixe S le seuil d'arrêt

2- Construction des partitions
- . on possède le dictionnaire $D_m = \{p_m^1, \cdots, p_m^k\}$ après m étapes.
- . on cherche une partition composée des "clusters" p_m^i:

$x \in p_m^i \Longleftrightarrow d(x, p_m^i) \leq d(x, p_m^j)$ pour $j = 1 \cdots k$
qui minimise l'erreur de quantification E_m associée à D_m:
$E_m = \frac{1}{N} \sum_{i=1}^{N} \underset{j}{Min}(d(x_i, p_m^j))$ pour $j = 1 \cdots k$
avec N : nombre d'élement d'apprentissage.

3- Test d'arrêt
- . $\frac{E_{m-1} - E_m}{E_m} \leq S$ alors on s'arrête
- . sinon aller en 4.

4- Recalcul des prototypes: Centroïdes
Le dictionnaire est désormais composé de nouveaux p_m^i.
- . on fait $m = m + 1$, aller en 2

2.3.2.2 Méthode de Linde, Buzo et Gray [Linde et al. 1980]: LBG

dans cette méthode chaque prototype p_m^i de rang m produit deux prototypes de rang $2m$:

$$p_m^i - \epsilon \text{ et } p_m^i + \epsilon$$

par l'intermédiaire d'un vecteur pertubation fixe ϵ.
L'algorithme général est le suivant :

1- Initialisation
. Fixer k le rang du dictionnaire (le nombre de prototypes) qui doit être une puissance de 2:
$k = 2^r$ où r est un entier.
. Fixer ϵ.
. Choisir le centre de gravité de l'ensemble d'apprentissage: p^0
. Faire $m = 0$

2- Eclatement : "Splitting"
. Tous les prototypes p^i du dictionnaire sont "éclatés" en $p^i - \epsilon$ et $p^i + \epsilon$
. Faire $m = m + 1$

3- Contruction des partitions
. Chercher les partitions autour de chaque prototypes.

4- Mise à jour des prototypes
.Recalculer les centroïdes de chaque partitions

5- Test d'arrêt
.Si $m \leq r$ aller en 2
.sinon arrêt

2.3.3 Les réseaux connexionnistes

2.3.3.1 Présentation

Les réseaux connexionnistes ou « réseaux de neurones » sont des méthodes numériques permettant de capturer des relations entre des événements ou des signaux caractéristiques d'un phénomène. Ils sont utilisés le plus souvent pour des problèmes pour lesquels on est capable de réaliser un grand nombre de mesures et dont on ne connaît pas la loi sous-jacente. Cela couvre une large gamme de

problèmes pouvant aller de la modélisation de phénomènes naturels à l'identification et la commande de processus industriels en passant par la reconnaissance des formes.
Le comportement d'un réseau sera caractérisé par un certain nombre de paramètres qui le définissent et qui sont déterminés à partir d'exemples de la tâche à réaliser par des algorithmes d'apprentissage. Ces derniers sont principalement basés sur des méthodes d'estimation statistique. Il existe une multiplicité de réseaux de neurones qui diffèrent par leur forme fonctionnelle, les classes de fonctions qu'ils permettent d'approximer, les critères qu'ils optimisent et les algorithmes d'apprentissage. Les réseaux de neurones ont notamment été utilisés pour des tâches de discrimination, de régression et d'approximation de fonctions. C'est ce qui nous intéressera dans cette thèse. Malgré la diversité des approches et des buts, la problématique de l'apprentissage est commune à l'ensemble de ces modèles. Dans la suite de brève présentation, nous considérerons deux modèles génériques qui sont le perceptron Multi-couches (PMC) [Rumelhart et al. 1986] et les réseaux à fonction de base radiales (RBF) [Poggio et Girosi1990], et nous renvoyons à [Hertz et al. 1991] pour une présentation plus détaillée des réseaux de neurones. Ces deux modèles sont les plus courants et les plus simples des réseaux non-linèaire. Ils sont basés sur un assemblage d'éléments appelés cellules. Il s'agit d'unités de calcul qui reçoivent une donnée en entrée, sous la forme d'un vecteur $\in \Re^n$, et produisent une sortie réelle. La fonction de transfert caractèrisant une telle unité est définie de $\Re^n$ dans $\Re$ et est de la forme :

$$y = f(A)$$

Où A est une fonction de $\Re^n$ dans $\Re$, et f une fonction $\Re$ dans $\Re$ (fonction de transition). Les versions de base de nos deux modèles sont les suivantes :

* PMC

$$A = w_0 + \sum_j w_j x_j$$

w_0 : biais

f : fonction $th(x)$ ou $(f(x) = \dfrac{1}{1+e^{-x}})$

* RBF

$$\text{couche de sortie: } A = w_0 + \sum_j w_j x_j$$

$$f = Id$$

$$\text{couche intermédiaire: } A = ||x - w||^2$$

$$f(x) = e^{-x}$$

La composition des fonctions de transition élémentaires des différentes unités est appelée fonction de transfert globale du réseau. Elle est définie de $\Re^n$ dans $\Re^m$. Pour la $i^{ème}$ sortie, cette fonction s'écrit :

- PMC à une couche cachée

$$y_i = \Psi_i(x) = f[w_{i0} + \sum_j w_{ij} f(w_{j0} + \sum_k w_{jk} x_k)]$$

où j décrit l'ensemble des unités cachées et k celles d'entrées.

- RBF

$$y_i = \Psi_i(x) = f[w_{i0} + \sum_j w_{ij} f(||x - w_j||^2)]$$

2.3.3.2 Critère d'apprentissage

L'apprentissage consiste à minimiser le risque empirique sur un ensemble d'apprentissage D_N de taille N :

$$R_{emp}(w) = \frac{1}{N} \sum_{k=1}^{N} [y^k - \Psi(x^k)]^2$$

Le modèle est entraîné à partir de D_N de façon à déterminer les meilleurs poids W^* qui minimisent $R_{emp}(w)$:

$$W^* = \underset{w}{argmin}(\frac{1}{N} \sum_{k=1}^{N} [y^k - \Psi(x^k)]^2)$$

Pour cela, on utilise généralement des procédures de gradient. Les versions les plus simples utilisent les dérivées premières de l'erreur, c'est le cas de l'algorithme dit de la plus grande pente :

$$w = w - \epsilon \frac{\partial R_{emp}(w)}{\partial w}$$

Où ϵ est le pas du gradient, il contrôle l'amplitude des modifications, et peut être fixe ou variable durant l'apprentissage.

2.4 Conclusion

L'un des points faibles du diagnostic à base de modèles réside dans le fait qu'il est très difficile de modéliser certains comportements normaux ou anormaux lorsque aucun modèle mathématique ne peut le décrire de façon simple. C'est

pourquoi nous avons choisit le monde de la reconnaissance des formes, et plus précisément les modèles connexionnistes. Leurs grandes capacités à modéliser à partir d'exemples des phénomènes très complexes est un atout majeur dans notre problème. En effet, il n'existe pas de modèles mathématiques pour décrire la fluctuation du trafic téléphonique, il nous fallait donc un système capable de le faire.

Chapitre 2. Bibliographie

[Boubour 1997] BOUBOUR (R.). – Suivi de pannes par corrélation causale d'alarmes dans les systèmes répartis : Application aux réseaux de télécommunication. *Thèse à l'université de Rennes 1, IRISA*, 1997.

[Dague et al. 1997] DAGUE (P.), GUERRIN (F.) et TRAVÉ-MASSUYÈS (L.). – *Le Raisonnement Qualitatif pour les sciences de l'ingenieur.* – HERMES, 1997.

[Dague 1994] DAGUE (P.). – Model-based diagnosis of analog electronic circuits. *Annals of Mathematics and Artificial Intelligence,special issue on Model-based Diagnosis, J.C. Baltzer*, vol. 11(1-4), 1994, pp. 439–492.

[Domart et Bourneuf 1981] DOMART (A.) et BOURNEUF (J.). – Nouveau Larousse Medical. *Librairie Larousse*, 1981.

[Dubuisson 1990] DUBUISSON (B.). – Diagnostic et reconnaissance des formes. *Traité des Nouvelles Technologies, série Diagnostic et Maintenance, Hermes*, 1990.

[Greiner et al. 1989] GREINER (R.), SMITH (B. A.) et WILKERSON (R. W.). – A correction to the algoithme in Reiter's theory of diagnosis. *Artificial Intelligence*, vol. 41(1), 1989, pp. 79–88.

[Hertz et al. 1991] HERTZ (J.), KROGH (A.) et PALMER (R.G.). – Introduction to the Theory of Neural Computation. *Addison Wesley*, vol. 1, 1991.

[Linde et al. 1980] LINDE (Y.), BUZO (A.) et GRAY (R.M.). – An Algorithm for the VQ Design. *IIIE, Trans. on Communication*, vol. 28, 1980, pp. 84–95.

[Macqueen 1967] MACQUEEN (J.). – Some Methods for Classification and Analysis of Multivariate Observations. *Fifth Berkeley Symposium on Mathematics, Statistics and Probabilities*, vol. 1, 1967, pp. 281–297.

[Poggio et Girosi 1990] POGGIO (T.) et GIROSI (F.). – Regularization algorithms that one equivalent to multilayer networks. *Science*, vol. 247, 1990, pp. 978–982.

[Poquelin 1666] POQUELIN (J-B.). – Le Médecin malgré lui. *Nouveaux classiques Larousse, Librairie Larousse 1971*, vol. Acte II, scéne IV, 1666, p. 57.

[Reiter 1987] REITER (R.). – A theory of diagnosis from first principales. *Artificial Intelligence*, vol. 32(1), 1987, pp. 57–96.

[Rumelhart et al. 1986] RUMELHART (D.E.), HINTON (G.E.) et WILLIAMS (R.J.). – Learning Internal Representations by Error Propagation. *Parallel Distributed Processing, MIT Press*, vol. 1, 1986.

Chapitre 3

Sélection et Extraction de Caractéristiques


e superflu finit
par priver du
nécessaire.

P.C. de LACLOS (1741-1803).

La sélection de variables est un problème de la statistique très ancien. Même si de nombreuses techniques ont été développées, il reste néanmoins ouvert. Dans ce chapitre, nous formalisons ce problème, et présentons différentes techniques classiques et connexionnistes parmis les plus rependues. Deux nouvelles mesures pour la sélection de variables: IIIE, et AINS, sont présentées. La première mesure est appliquée sur des modèles de type PMC et utilisée pour l'extraction de règles symbolique à partir de modèles connexionnistes. Notre seconde mesure a été développée afin de permettre la sélection de variables sur d'autres modèles connexionnistes que ceux de type PMC. Nous validons cette méthode, non seulement sur des PMCs, mais aussi sur des modèles de type RBF. Pour finir, nous terminons ce chapitre par la comparaison de nos deux mesures à différentes techniques connexionnistes.

3.1 Sélection de variables

Dans les tâches de modélisation et de discrimination, la quantité et la qualité d'information sont essentielles. On pourrait même se dire que plus on disposera d'un ensemble d'information important, plus le système que nous concevrons sera performant et fiable. Il faut nuancer ce type d'idée. Si l'information est vitale pour de tels systèmes, il s'avère que la redondance d'information (a fortiori superflue) nuit au bon fonctionnement de ceux-ci et ajoute de surcroît du calcul. Font partie des informations "néfastes" celles qui n'apportent rien à la définition du phénomène étudié, comme une variable restant constante ou bien n'ayant aucune corrélation avec celui-ci. Il faut y ajouter les variables informatives mais exagérément bruitées et qui par le fait qu'elles sont trop bruitées, rendent leur utilisation impropre. On peut classifier les techniques mathématiques de sélection de variables en deux catégories :

- La sélection de variables dans l'espace des données.
- La sélection dans un espace transformé: extraction de caractéristiques.

La première catégorie a pour but de chercher d' variables parmi les d originelles. La deuxième consiste à définir de nouvelles variables à partir de l'ensemble des variables originelles. Une technique simple consiste à faire une ACP (Analyse en Composantes Principales) sur les échantillons et prendre comme nouvelles variables les p composantes principales. Ces nouvelles caractéristiques sont calculées sous forme de combinaisons linéaires des variables originelles. Ce bref parcours nous montre l'utilité d'une bonne sélection de variables. En résumé, la sélection de variables nécessite généralement trois éléments essentiels [Bennani1998] qui sont :

- un critère J d'évaluation de l'importance:
  Il s'agit de déterminer, l'apport d'une variable (ou d'un sous ensemble de variables) dans la qualité de discrimination d'un système, ou lorsqu'il s'agit d'un problème de régression, la qualité de prédiction.
- une procédure de recherche d'un sous ensemble de variables :
  Dans le cas géneral les mesures de pertinence d'une variable utilisée ne permettent pas d'obtenir un ensemble ordonné, permettant d'obtenir le sous-ensemble optimal. Seule une recherche exhaustive (pour p variables $2^p - 1$ combinaisons) en est capable, ce qui est souvent irréalisable. C'est pourquoi, les procédures de recherche sont nécessaires. On peut alors soit faire appel à une méthode d'optimisation de type Branch an Bound [De bruin et al. 1988]. Soit avoir recours à des méthodes sous-optimales.

Les procédures sous-optimales de sélections les plus simples sont des procédures séquentielles comme:

- la sélection ascendante
- l'élimination descendante
- la "stepwise selection"

Soit C l'ensemble des variables, C_k l'ensemble des variables sélectionnées à la $k^{\text{ème}}$ étape et $v_1, v_2, \cdots, v_{d-k}$ les variables encore diponibles.
Pour la sélection ascendante, on retient la variables v_i telle que :

$$J(C_k) = \underset{v_i \in C - C_{k-1}}{Max} J(C_{k-1} \cup \{v_i\})$$

au départ, l'ensemble des variables est vide.
La procédure d'élimanation descendante est inverse à la précedente. On part de l'ensemble complet de toutes les variables. A l'étape k, on ôte la variable v_i telle que:

$$J(C_k) = \underset{v_i \in C_{k+1}}{Max} J(C_{k+1} - \{v_i\})$$

Pour la méthode "stepwise selection" le principe général consiste á sélectionner les variables une à une et à chaque étape on teste à nouveau si une d'elles ne peut-être éliminée.

- un critère d'arrêt de la procédure de recherche :
  Dans de nombreux cas, le nombre de variables à sélectionner n'est pas connu a priori. Il faut donc déterminer à qu'elle moment une variable n'est plus considérée informative.

3.2 Méthodes Statistiques

La sélection de variables a été largement étudiée dans la littérature statistique de la reconnaissance des formes [Fukunaga1990].
Nous rappelons dans cette section quelques méthodes de sélection de variables. Ces méthodes peuvent être utilisées comme méthodes de réference.

3.2.1 Critères probabilistes de sélection

En discrimination, le but est d'effectuer la reconnaissance avec le moins d'erreurs possible. Ainsi, la sélection de variables est faite de façon à conserver la meilleure séparabilité des classes.
La distance probabiliste entre les fonctions de densités des classes, $p(x|w_i)$ et $p(x|w_j)$ peut servir comme critère de séparabilité des classes w_i et w_j. Plus cette

distance est petite, plus l'erreur est grande et inversement. Le problème majeur de ces critères est leur calculabilité puisqu'en général on ne connait pas leur expression analytique.

3.2.2 Critères déterministes de sélection

Les mesures de distance inter-classes, sont utilisées pour quantifier l'importance des ensembles de variables. Ce sont les critères les plus utilisés.
Par exemple nous pouvons définir les matrices de distance intra-classe W et inter-classe B.
L'erreur de discrimination est minimale si la distance inter-classe est plus grande et la distance intra-classe plus petite. On maximise alors le critère heuristique suivant:

$$J = \frac{|W + B|}{|W|}$$

où $|.|$ ici note le déterminant des matrices.
Il existe plusieurs critères exprimés en fonction des traces des matrices ou des critères non linéaires qui s'appuient sur l'estimation de fenêtres de Parzen ou sur la mesure de Patrick-Fisher [Derijver et Kittler1982].

3.3 Méthodes Connexionnistes

La sélection de variables connexionnistes est une approche différente, car le nombre de variables est directement lié à l'architecture du modèle et au nombre de ses paramètres (la complexité de la fonction calculable par le réseau). Dans le langage connexionniste la sélection de variables correspond à une technique d'élagage de variables.
Les méthodes connexionnistes de sélection de variables se basent en général sur des critères heuristiques permettant d'estimer l'importance d'une ou de plusieurs variables sur la performance globale du système.
Les méthodes connexionnistes de sélection de variables sont en général de type "backward". Elles se basent sur l'élimination successive des variables les moins importantes. L'utilisation d'un nombre différent de variables doit être suivie par un réapprentissage du réseau connexionniste.
Nous allons passer maintenant en revue quelques méthodes connexionnistes représentatives de sélection de variables.

3.3.1 HVS [Yacoub et Bennani1997]

Yacoub et Bennani proposent une mesure de pertinence basée sur la valeur des poids et la structure du réseau.

Dans le cas d'un PMC à une couche cachée cette mesure est définie par:

$$pertinence(x_i) = \sum_{j \in H} \left(\frac{|w_{ji}|}{\sum_{i' \in I} |w_{ji'}|} \sum_{k \in O} \frac{|w_{kj}|}{\sum_{j' \in H} |w_{kj'}|} \right)$$

où I est la couche d'entrée, H la couche cachée et O la couche de sortie.
Cette mesure peut se généraliser dans le cas d'un PMC à plusieurs couches cachées.

3.3.2 Saliency Based Pruning (SBP) [Moody et Utans1992], [Moody1994]

Cette méthode utilise une mesure de pertinence basée sur la variation de l'erreur en apprentissage lorsqu'on remplace une variable x_i par sa moyenne $\overline{x}_i$.

$$pertinence(x_i) = MSE - MSE(\overline{x}_i)$$

avec MSE: erreur quadratique moyenne.

3.3.3 Méthodes basées sur les dérivées des sorties du réseau [Hashem1992]

La dérivée de la fonction F que représente le réseau par rapport à chacune des variables x_i est souvent utilisée comme mesure de pertinence.

$$pertinence(x_i) = \frac{\partial F}{\partial x}$$

Plusieurs mesures de ce type on été proposées. Par exemples [Ruck et al. 1990] proposent la mesure de pertinence suivante:

$$pertinence(x_i) = \sum_{p=1}^{N} \sum_{s=1}^{m} \left| \frac{\partial F_s}{\partial x_i}(x^p) \right|$$

où N est la taille de la base d'apprentissage, m le nombre de variables en sortie, x^p le $p^{\text{ème}}$ exemple et x_i la $i^{\text{ème}}$ variable.

3.3.4 OCD: Optimal Cell Damage [Cibas et al. 1994]

Cette méthode est une extension de la technique d'élagage OBD (Optimal Brain Damage) proposée par Yann Le Cun [Le cun et al. 1990].

La méthode OCD utilise les informations fournies par OBD pour définir la pertinence d'une variable comme la somme des pertinences des connexions qui partent de celle-ci.

$$pertinence(x_i) = \sum_{j \in fan\text{-}out(x_i)} pertinence(w_{ji})$$

où $fan\text{-}out(x_i)$ représente l'ensemble des poids partant de la variable x_i.
Dans OBD, la pertinence d'une connexion est définie par:

$$pertinence(w_{ji}) = \frac{1}{2}\frac{\partial^2 MSE}{\partial w_{ji}^2} w_{ji}^2$$

Dans OCD:

$$pertinence(x_i) = \sum_{j \in fan\text{-}out(x_i)} \frac{1}{2}\frac{\partial^2 MSE}{\partial w_{ji}^2} w_{ji}^2$$

Une variante a été proposée par Leray et Gallinari [Leray1998] utilisant les informations fournies par EBD (Early Brain Damage) [Tresp et al. 1997]. EBD est une extension de OBD. Dans ce cas la pertinence de la variable x_i est définie par:

$$pertinence(x_i) = \sum_{j \in fan\text{-}out(x_i)} \frac{1}{2}\frac{\partial^2 MSE}{\partial w_{ji}^2} w_{ji}^2 - \frac{1}{2}\frac{\partial MSE}{\partial w_{ji}} w_{ji} + \frac{1}{2}\frac{(\frac{\partial MSE}{\partial w_{ji}})^2}{\frac{\partial^2 MSE}{\partial w_{ji}^2}}$$

3.4 Deux nouvelles mesures de pertinence

Dans cette section, nous décrivons deux nouvelles méthodes de sélection de variables relatives aux systèmes connexionnistes. Ces nouvelles méthodes identifient et sélectionnent les variables importantes, c'est à dire caractéristiques du problème.
Ceci permet de supprimer les variables redondantes ainsi que les variables ne contenant pas d'information caractéristique du problème. Ces approches combinent deux catégories de technique de sélection de variables : sélection et extraction de traits.
Elles sont basées sur le calcul de l'effet d'un neurone d'entrée sur la sortie d'un réseau de neurones. Ces techniques sont capables de caractériser l'information importante contenue dans chaque exemple présenté au réseau.

3.4.1 IIIE (de l'Information Implicite à l'Information Explicite)

3.4.1.1 L'idée

Nous proposons de déterminer l'effet (Degré de participation) qu'a un neurone d'entrée sur un neurone de sortie. Cet effet est calculé de la façon suivante :

- Pour chaque neurone de la couche cachée, nous déterminons sa participation sur l'activation d'un neurone de sortie
- Nous calculons la participation de chaque neurone d'entrée sur l'activation d'un neurone de la couche cachée
- On en déduit la participation d'un neurone d'entrée sur l'activation d'un neurone de la couche de sortie via un neurone de la couche cachée.
- Nous sommons les participations pour tous les neurones de la couche cachée, nous obtenons ainsi la participation d'un neurone d'entrée sur un neurone de sortie.

Avant de développer le calcul de notre approche sur l'importance des neurones, nous allons introduire quelques notations et définitions.

3.4.1.2 Mesure de pertinence

Notations et définitions

f : Fonction de transition (Sigmoïde).

w_{ij} : Poids de la connexion du neurone j au neurone i.

$A_i = \sum_j x_j w_{ij}$: Activation du neurone i (Somme pondérée).

$x_i = f(A_i)$: État du neurone i.

x_i^p : État d'un neurone i de la couche d'entrée, pour un exemple p

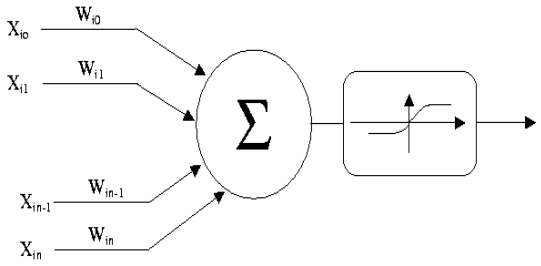

FIG. 3.1: *Schéma d'un neurone.*

Décomposition de l'activation d'un neurone

Pour simplifier, nous considérerons un réseau connexionniste à trois couches. Nous décomposons l'entrée d'un neurone i en deux termes A_i^N et A_i^C, de la façon suivante :

$$A_i = A_i^C + A_i^N \tag{3.1}$$

A_i^C est définie comme la partie *coopérative* de l'entrée du neurone i, c'est dire la quantité :

$$A_i^C = \sum_j x_j w_{ij} \text{ où la somme sur j est telle que } x_j w_{ij} A_i \geq 0 \tag{3.2}$$

C'est-à-dire que A_i^C est la quantité de même signe que l'activation du neurone i. Au contraire A_i^N est définie comme la partie *non coopérative* de l'activation du neurone i, c'est à dire la quantité :

$$A_i^N = \sum_j x_j w_{ij} \text{ où la somme sur j est telle que } x_j w_{ij} A_i < 0 \tag{3.3}$$

C'est à dire que A_i^N est la quantité de signe contraire à l'entrée du neurone i.

Décomposition de l'état d'un neurone

Nous décomposons de la même manière l'état x_i d'un neurone i, en deux parties x_i^C et x_i^N, où les quantités x_i^C et x_i^N sont définies comme suit:

$$\text{posons } \psi(a,b) = \int_a^b f(x)\,dx, \tag{3.4}$$

on estime la variation causée par la partie coopérative (A_i^C) de l'état d'un neurone par :

$$\Delta_i^C = \psi(0, A_i^C) \tag{3.5}$$

et la variation de l'état d'un neurone causée par la partie non-coopérative (A_i^N) par:

$$\Delta_i^N = \psi(0, A_i) - \psi(0, A_i^C) \tag{3.6}$$

On définit la partie coopérative de l'état d'un neurone comme:

$$x_i^C = (1 - \tau)x_i \tag{3.7}$$

la partie non-coopérative comme:

$$x_i^N = \tau x_i \tag{3.8}$$

avec $\tau = \frac{\Delta_i^N}{\Delta_i^C}$, qui représente la proportion de variation causée par A_i^C relativement à celle causée par A_i^N dans le calcul de l'état du neurone i.

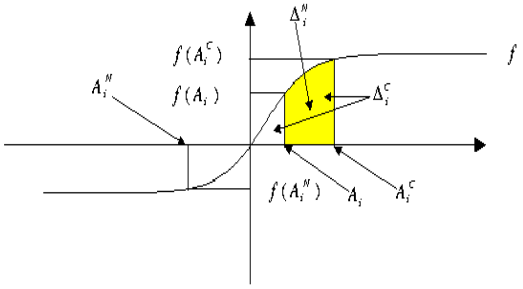

FIG. 3.2: *Calcul d'influence.*

Recherche de l'effet d'une entrée sur la sortie

– *Pour l'entrée et la couche cachée:*
Le degré de participation d'un neurone d'entrée i sur un neurone caché j, est défini comme suit :

$$\Pi_{ij} = \begin{cases} \frac{x_i w_{ji}}{A_j^C} x_j^C & \text{si } x_i w_{ji} A_j \geq 0 \\ \frac{x_i w_{ji}}{A_j^N} x_j^N & \text{si } x_i w_{ji} A_j < 0 \end{cases} \tag{3.9}$$

Pour essayer de donner une interprétation intuitive de cette formule, on peut voir le terme $\frac{x_i w_{ji}}{A_j^C}$ comme représentant la proportion de la contribution du neurone i dans A_j^C (partie coopérative de A_j).

Cette proportion est utilisée comme un facteur pondérant de x_j^C pour donner finalement la partie coopérative du neurone i dans l'état x_j du neurone j.
On peut appliquer le même raisonnement pour la seconde ligne de la formule 3.9, pour la partie non-coopérative.

– *De l'entrée à la couche de sortie via la couche cachée:*
Le degré de participation d'un neurone d'entrée i sur un neurone de sortie k via un neurone de la couche cachée j, pour un exemple p, est défini par δ_{ijk} comme suit:

$$\delta_{ijk} = \begin{cases} \frac{\Pi_{ij} w_{kj}}{A_k^C} x_k^C & \text{si } \Pi_{ij} w_{kj} A_k \geq 0 \\ \frac{\Pi_{ij} w_{kj}}{A_k^N} x_k^N & \text{si } \Pi_{ij} w_{kj} A_k < 0 \end{cases} \tag{3.10}$$

Une interprétation intuitive de cette formule peut être vue comme suit : pour cette formule nous utilisons la participation d'un neurone d'entrée i sur un neurone caché j pour estimer la proportion de la contribution sur le neurone k dans A_k^C représenté par le terme $\frac{\Pi_{ij} w_{kj}}{A_k^C}$.
Cette proportion est utilisée comme un facteur pondérant de x_k^C pour donner finalement la partie coopérative du neurone i dans l'état x_k du neurone k via le neurone j.
On peut appliquer le même raisonnement pour la seconde ligne de la formule 3.10, pour la partie non-coopérative.

– *De l'entrée à la sortie :*
Le degré total de participation d'un neurone d'entrée i sur le neurone de sortie k est donné de la façon suivante :

$$\gamma_{ik} = \sum_{j \in fan\text{-}in(k)} \delta_{ijk} \tag{3.11}$$

Dans la tâche de classification pour un réseau de c neurones de sortie, la décision du système est basée sur la cellule la plus active lorsqu'on présente un exemple p de classe s.

L'importance d'un neurone d'entrée i sur une décision s du sytème possédant n_o neurones de sorties (classes) pour un exemple donnée de classe c, sera définie par:

$$pertinence(x_i^p) = (n_o - 1)\gamma_{ic} - \sum_{k \in O, k \neq c} \gamma_{ik} \tag{3.12}$$

où O est la couche de sortie.
Il faut noter que nous transformerons très souvant ces pertinence sous forme de pourcentage, de la manière suivante :

$$pertinence(x_i^p) = \frac{pertinence(x_i^p)}{\sum_j pertinence(x_j^p)}$$

Ceci implique, qu'il nous arrivera de définir des pourcentages négatifs, cela se trovant être très utile lorque nous utilisons cette mèthode pour l'extraction de règles symboliques. La valeur negative nous informant qu'il faut prendre comme pertinent la valeur complementaire de la variable concernée (quand celle-ci est une variable "booléenne").
Il est intéressant de définir la complexité de cet algorithme de sélection de variables dans le cas général.
Si n est le nombre de neurones d'entrée, C_i le nombre de neurone d'une couche cachée i, S le nombre de neurones de sortie et k le nombre de couches cachées alors la complexité est la suivante:

- De l'entrée à la première couche cachée : nC_1
- D'une couche cachée i à une couche cachée $i+1$: nC_iC_{i+1}
- De la dernière couche cachée C_k à la couche de sortie: nC_kS

La complexité globale est donc:

$$nC_1 + \sum_j nC_jCj + 1 + nC_kS$$

3.4.1.3 Procédure de recherche et critère d'arrêt

On utilise un réseau connexionniste entraîné. Pour chaque exemple, on calcule le degré de participation d'un neurone d'entrée sur la sortie du réseau.
En utilisant la formule 3.12, pour un exemple p de la classe s, on calcule pour chaque neurones d'entrée i $pertinence(x_i^p)$. Après présentation des exemples de la base d'apprentissage, on calcule l'importance moyenne pour chaque neurone d'entrée i. Les variables sont triées selon l'ordre croissant de pertinence et la variable de plus faible importance est alors supprimée.
Ce principe génère un ensemble de k réseaux (k étant le nombre de variables du réseau) ayant de moins en moins de variables.
Soit $PMC(i), i = 1, \cdots, k$ ces réseaux et $E(i)$ l'erreur estimée sur une base de validation.
A ce niveau deux méthodes sont possibles pour la sélection d'un sous-ensemble de variables:

- La première consiste à choisir le réseau $PMC(i^*)$ qui offre la meilleure performance:

$$PMC(i^*) = \underset{PMC(i)}{argmin} E(i)$$

et sélectionner le sous-ensemble des i^* variables du $PMC(i^*)$.

- la deuxième méthode consiste à utiliser un test statistique (test de Fisher) et chercher le réseau statistiquement proche du meilleur réseau en performance et possédant le plus faible nombre de variables.

3.4.2 AINS (Architecture Independent Neural Selection)

Nous avons montré dans les paragraphes précédents l'utilité essentielle de la sélection de variables dans les performances des PMCs.
Il s'avère qu'il en va de même pour les modèles connexionnistes de type RBF.
La majorité des techniques de sélection de variables que l'on peut trouver dans le domaine s'appliquent principalement pour des modèles de type PMC. Nous proposons dans ce paragraphe une approche indépendante du modèle connexionniste utilisé. Celle-ci sera validée sur les deux modèles connexionnistes les plus utilisés à savoir PMC et RBF. On doit noter que cette technique, outre la possibilité de sélection de variables, comme pour IIIE, permet en natif d'extraire des règles et d'optimiser l'architecture (perspective que nous comptons mettre en oeuvre).

3.4.2.1 Mesure de pertinence

Le but dans ce paragraphe est de développer une mesure permettant de déterminer la contribution d'une variable d'entrée sur la valeur X d'une fonction f non linéaire. Soit $X = f(X_1 + X_2 + \cdots + X_n)$ avec $X \in \Re$, $\forall i X_i \in \Re$ et f une fonction non linéaire. La contribution du terme X_i sur la valeur X sera définie par $\Pi(X_i, X)$. Cette contribution est supposée de la forme suivante:

$$\Pi(X_i, X) = f(X_i) + \varepsilon(X_1, \cdots, X_n) \tag{3.13}$$

$$\text{telle que: } \sum_{i=1}^{n} \Pi(X_i, X) = X \tag{3.14}$$

$\varepsilon(X_1, \cdots, X_n)$ est un terme à déterminer.
Les deux équations précédentes donnent:

$$\sum_{i=1}^{n}[f(X_i) + \varepsilon(X_1, \cdots, X_n)] = X \tag{3.15}$$

$$\sum_{i=1}^{n} f(X_i) + n\varepsilon(X_1, \cdots, X_n) = X \tag{3.16}$$

$$\Longrightarrow \varepsilon(X_1, \cdots, X_n) = \frac{X - \sum_{i=1}^{n} f(X_i)}{n} \tag{3.17}$$

$$\text{d'où } \Pi(X_i, X) = f(X_i) + \frac{X - \sum_{i=1}^{n} f(X_i)}{n} \tag{3.18}$$

Application aux modèles connexionnistes

Les réseaux de neurones artificiels sont composés de fonctions sommantes à plusieurs entrées. Nous avons donc appliqué notre mesure $\Pi(X_i, X)$ aux modèles connexionnistes. Le résultat de notre mesure pour ces modèles sera noté AINS(N_i, N_j) où N_i et N_j sont deux neurones connectés directement ou indirectement.

Notre approche est la suivante:
Nous définissons les différentes caractéristiques d'un neurone N_i comme suit:

- f_i La fonction de transition d'un neurone N_i (c.à.d. une fonction sigmoïde)
- w_{ij} le poids de la connexion du neurone N_j au neurone N_i
- $\Omega_i(x_j, w_{ij})$ le *somma* calculé pour le neurone N_i utilisant les valeurs venant du neurone N_j (typiquement $x_j.w_{ij}$ pour les unités scalaires et $(x_j - w_{ij})^2$ pour les unités euclidiennes.)
- $x_i = f_i(\sum_j \Omega_i(x_j, w_{ij}))$, la sortie calculée pour le neurone N_i
- $fan\text{-}in(N_j)$ l'ensemble des neurones connectés directement au neurone N_j

Pour expliquer notre méthode, nous avons utilisé les deux exemples suivants (Cf Fig. 3.3). Nous considérons deux réseaux l'un avec couche cachée et l'autre sans couche cachée.

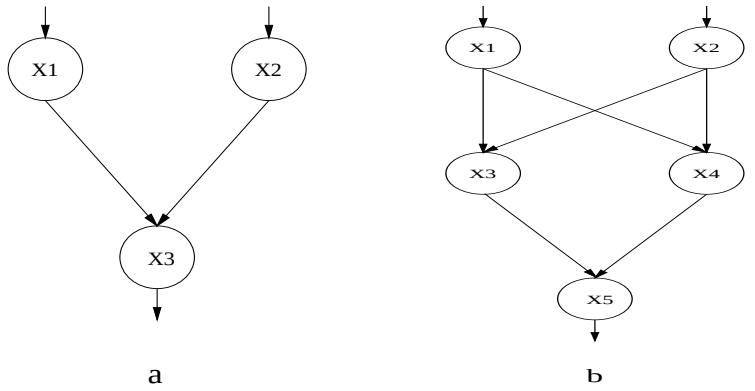

FIG. 3.3: *Exemple de base du calcul de AINS sur la sortie d'un réseau sans couche cachée (a), et avec couche cachée (b)*

- Pour le réseau sans couche cachée.
  La mesure AINS(N_1, N_3) (c.à.d. la contribution du neurone N_1 sur N_3) est simplement l'application directe de Π au calcul de l'activation du neurone N_3. Elle est définie comme ceci :

$$\text{AINS}(N_1, N_3) = f_3(\Omega_3(N_1, w_{31})) + \frac{N_3 - \sum_{k\in(1,2)} f_3(\Omega_3(N_k, w_{3k}))}{2} \tag{3.19}$$

- Pour le réseau avec couche cachée .
  La mesure AINS(N_1, N_5) doit être calculée via tous les neurones connectés entre les neurones N_1 et N_5. Nous obtenons donc la formule récursive suivante :

$$\begin{aligned}\text{AINS}(N_1, N_5) = {} & f_5(\Omega_5(\text{AINS}(N_1, N_3), w_{53}) + \Omega_5(\text{AINS}(N_1, N_4), w_{54})) \\ & + \frac{x_5 - \sum_{k\in(1,2)} f_5(\Omega_5(\text{AINS}(N_k, N_3), w_{53}) + \Omega_5(\text{AINS}(N_k, N_4), w_{54}))}{2}\end{aligned} \tag{3.20}$$

Nous aboutissons à la formule récursive générale suivante :

- $\forall N_i \in$ input
- $\forall N_j \in$ output
- Cas de base :

$$\text{AINS}(N_i, N_i) = x_i \tag{3.21}$$

– Cas général :

$$\mathrm{AINS}(N_i, N_j) = f_j\Big(\sum_{k \in fan\text{-}in(N_j)} \Omega_j(\mathrm{AINS}(N_i, N_k), w_{jk})\Big) + \frac{x_j - \sum_{k \in \mathrm{input}} f_j\Big(\sum_{t \in fan\text{-}in(N_j)} \Omega_j(\mathrm{AINS}(N_k, N_t), w_{jt})\Big)}{\mathrm{length(input)}} \tag{3.22}$$

Pour les tâches de classification, nous avons considéré qu'une variable serait coopérative sur une sortie, si la mesure de pertinence est de même signe que la sortie, et non coopérative sinon. Le calcul de la complexité est le même que pour IIIE à savoir: Quand n est le nombre de neurones d'entrée, C_i le nombre de neurone d'une couche cachée i, S le nombre de neurones de sortie et k le nombre de couches diffétentes de la couche de sortie et de la couche d'entrée alors la complexité est la suivante:

– De l'entrée à la première couche: nC_1

– D'une couche cachée i à une couche cachée $i+1$: nC_iC_{i+1}

– De la dernière couche C_k à la couche de sortie: nC_kS

La complexité globale est donc:

$$nC_1 + \sum_j nC_jCj + 1 + nC_kS$$

3.4.2.2 Procédure de recherche et critère d'arrêt

Le problème lorsque l'on dispose d'une mesure reflétant l'importance d'un neurone dans une architecture connexionniste, est de déterminer si la valeur de cette mesure doit impliquer la suppression du neurone ou non. Pour pallier ce problème, ayant un ordre sur les neurones définis par notre méthode, nous avons classé par ordre croissant des influences l'ensemble des neurones. Pour notre expérimentation nous avons donc utilisé l'algorithme suivant :

1. Entraîner le réseau sur la base d'apprentissage, jusqu'à obtenir le minimum sur la base de validation (c.à.d. l'utilisation du *early-stopping*).

2. Calculer la contribution de chaque neurone d'entrée à l'aide de AINS et ainsi définir la liste triée L des neurones.

3. Soit D_0^v=performance courante du réseau sur la base de validation.
4. Sauvegarder le réseau.
5. Supprimer le neurone ayant la plus faible mesure de contribution dans L.
6. Réentraîner le réseau, jusqu'à obtenir le minimum sur la base de validation.
7. Si $D_0^v \leq$performance courante du réseau sur la base de validation, aller en 3.

La meilleure architecture est donnée par le dernier réseau sauvegardé.
Le critère d'arrêt est du même type que IIIE, on conserve l'ensemble des variables ayant maximisé les performances du système.

3.5 Validation de IIIE

Dans cette validation, il faut noter que le nombre de variables supprimées a été determiné par validation crois'e, c'est-à-dire que nous avons conservé le réseau donnant les meilleures performances en validation.

3.5.1 *ou-exclusif* bruité

Le problème du *ou-exclusif* a été identifié par Minsky et Papert comme une fonction booléenne simple, non séparable linéairement.
Cette fonction a été utilisée comme problème jouet. Nous avons ajouté à cette fonction du bruit de la manière suivante: P_1, P_2, P_3, P_4 quatre variables booléennes, Q est la fonction définie par $P_2 \bigotimes P_4$, P_1 et P_3 jouent le rôle de bruit.

Conditions expérimentales
La base d'apprentissage a été définie par la table de vérité de Q. L'idée de ce test est de vérifier que le réseau de neurones est capable de retrouver Q malgré le bruit.
Pour les trois problèmes de validation, nous avons utilisé des perceptrons multicouches entièrement connectés à trois couches.
Ces architectures de réseau ont une intéressante capacité d'analyse de problème. Durant l'apprentissage, les cellules de la couche cachée déterminent comment extraire les traits significatifs du signal d'entrée.
Les notations que nous avons utilisées pour représenter les architectures sont les suivantes :
$< in|hid|out >$, où in représente la dimension de la couche d'entrée, hid le nombre de neurones cachés et out le nombre de neurones de la couche de sortie.
Pour le problème du *ou-exclusif* bruité, l'architecture utilisée est $< 4|2|2 >$.

Résultats et interprétations
La figure 3.4 donne les résultats pour le problème du *ou-exclusif* bruité.
Cette figure montre l'influence des variables sélectionnées sur l'ensemble d'apprentissage, les parties en noir correspondent aux variables supprimées.

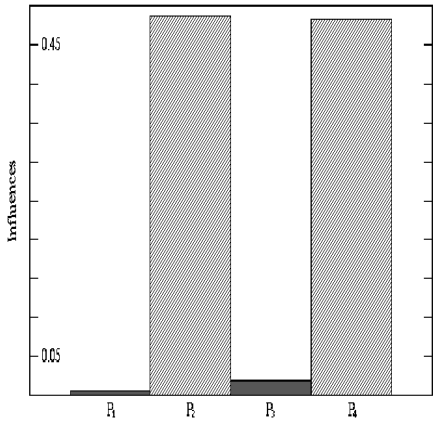

FIG. 3.4: *Influence des variables.*

Pour ce problème, les variables sélectionnées montrent clairement que le réseau connexionniste a bien détecté la position des variables bruitées (P_1 et P_3). En effet, la détection est évidente au vu des fortes influences de (P_2 et P_4) et de celles extrêmement faibles des variables bruitées (P_1 et P_3).

3.5.2 Le réseau téléphonique

Ce problème est extrait du travail de diagnostic que nous avons effectué dans cette thèse. C'est un problème à cinq classes, on dispose de 18 indicateurs du trafic téléphonique pour un instant t (cf tableau 3.1), et le but est d'apprendre au réseau connexionniste, l'identification à partir de 18 indicateurs de l'instant $t-1$ et 18 de l'instant $t-2$, l'état du réseau téléphonique à l'instant $t+1$. Nous avons utilisé comme architecture du réseau $< 36|20|5 >$.

1	Appels offerts	2	Appels offerts origine
3	Appels offerts destination	4	Appels échec distant
5	Prises efficaces	6	Prises efficaces origine
7	Prises efficaces destination	8	Appels efficaces
9	Appels efficaces origine	10	Appels efficaces destination
11	Trafic écoulé	12	Trafic écoulé origine
13	Trafic écoulé destination	14	Trafic efficace
15	Trafic efficace origine	16	Trafic efficace destination
17	Taux d'occupation	18	Taux d'efficacité

TAB. 3.1: *Indicateur du trafic téléphonique.*

Conditions expérimentales

Nous avons une base d'apprentissage de 4000 exemples, qui correspondent aux différents états du réseau, c'est à dire 800 exemples par état. La base de validation et de test sont toutes deux composées de 2000 exemples. On peut voir sur la figure 6.7 l'évolution de ces indicateurs pour une journée en état nominal. L'architecture utilisée est $< 36|20|5 >$.

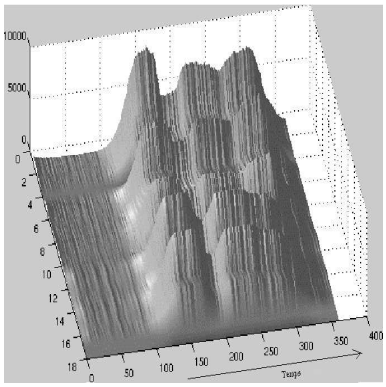

FIG. 3.5: *Évolution temporale et spatiale des indicateurs pour la situation nominale.*

Résultats et interprétations

Pour la sélection des indicateurs du réseau téléphonique, nous avons sur la figure 3.6, l'influence des 36 indicateurs, celle-ci montre les 31 indicateurs pertinants. Le tableau 3.2 montre le gain qu'apporte notre sélection de variables.

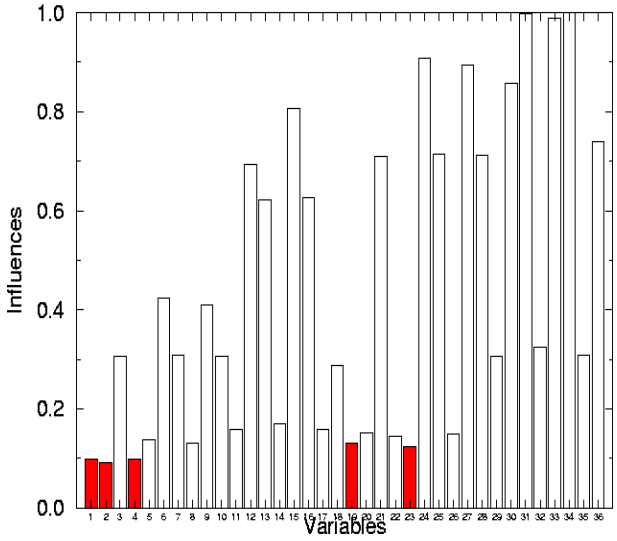

FIG. 3.6: *Influence des variables pour le problème du réseau téléphonique.*

Avant sélection $< 36\|20\|5 >$	Après sélection $< 31\|20\|5 >$
86.55% $\pm 1.57\%$	87.30% $\pm 1.53\%$

TAB. 3.2: *Résultat expérimentaux sur le réseau téléphonique.*

Dans le tableau 3.3, on peut noter que les indicateurs non sélectionnés sont le plus souvant des combinaisons d'autres indicateurs.

1	Appels offerts	aux temps $t-1$ et $t-2$
2	Appels offerts origine	au temps $t-1$
4	Appels échec distant	au temps $t-1$
5	Prises efficaces	au temps $t-2$

TAB. 3.3: *Indicateurs non sélectionnés du trafic téléphonique.*

3.5.3 Les vagues de Breiman

Ce problème a été proposé par Breiman dans [Breiman et al. 1984] (cf. Fig. 3.7) et une version bruitée a été utilisée par De Bollivier [Bollivier et al. 1991]. C'est un problème à trois classes générées par les vagues.
Les exemples sont des vecteurs de dimensions 40 construits comme des combinaisons convexes de deux vagues bruitées, nous présentons ci-dessous (cf équation 3.23) la composition de ces vecteurs. l'architecture utilisée ici est $< 40|10|3 >$

Soit x un vecteur forme. Sa $i^{ème}$ composante est définie par:

$$x_i = \begin{cases} \frac{[uh_i^m + (1-u)h_i^m]}{5} + \epsilon_i \text{ si } 0 \leq i \leq 20 \\ \epsilon_i \text{ si } 21 \leq i \leq 39 \end{cases} \tag{3.23}$$

avec :
$u \in \{0, 1\}$
Classe 0: $n = 1, m = 2$
Classe 1: $n = 1, m = 3$
Classe 2: $n = 2, m = 3$

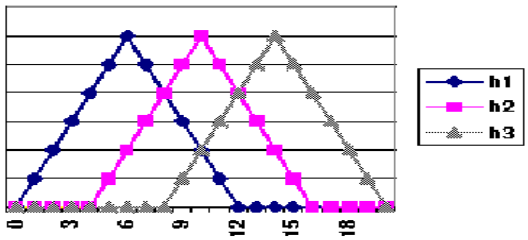

FIG. 3.7: *Vagues de Breiman.*

Conditions expérimentales
le protocole d'expérimentation a été le suivant :

- Pour la base d'apprentissage D_a nous avons utilisé 300 instances et 700 instances pour D_v, l'ensemble de la base de validation;
- Pour le test, nous avons utilisé l'ensemble D_t de 4300 instances;

Résultats et interprétations
Pour le problème des vagues de Breiman, notre méthode de sélection de variables a donné l'ensemble θ={11, 14, 7, 15, 13, 6, 5, 9, 8, 16, 18, 17, 12, 4, 10, 19 } qui est l'ensemble des variables les plus importantes.
Les éléments de cet ensemble sont donnés par ordre décroissant d'influence. La figure 3.8 donne l'influence de toutes les variables, ces valeurs sont réarrangées entre 0 et 1. Sur cette figure, on peut noter que les variables les plus importantes sont parmi les premières composantes, à l'exception des variables 1, 2, 3 qui sont rejetées.
De même, on remarque que toutes les composantes qui jouaient le rôle de bruit ont été éliminées.

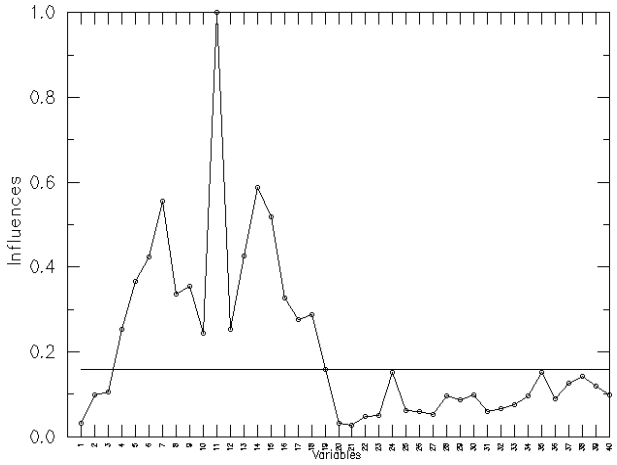

FIG. 3.8: *Influence des variables.*

Réseau connexionniste avec 10 neurones cachés: $< V - 10 - 3 >$	Da	Dv	Dt
Toutes les variables V=40	92 % ± 3.63%	83.86 % ± 2.90%	82.84 % ± 1.15%
Variables sélectionnées V=16	88.67 % ± 4.08%	85.57 % ± 2.80%	85.37 % ± 1.09%

TAB. 3.4: *Performances sur les "vagues de Breiman".*

Le tableau 3.4 nous donne les performances pour l'architecture $< 40|10|3 >$ (ici le réseau connexionniste non réentraîné après la sélection de variables donne les mêmes performances que dans le cas où on le réentraîne). Les résultats montrent qu'après suppression des variables, les performances sont bien meilleures. En effet, l'augmentation est de 2.53% sur la base de test. Ces dernières valeurs sont très proches de la limite théorique de bonne classification, qui est de 86 %.

3.5.3.1 Conclusion

Dans cette section, nous avons mis en évidence l'utilité de la sélection de variables à l'aide de la mesure IIIE sur les performances de modèles connexionnistes. Cette approche peut être étendue à d'autres modèles de type PMC et permet l'optimisation d'architecture en se basant sur l'influence des neurones de la couche cachée. Nous détaillerons ce point dans les perspectives.

3.6 Validation de AINS sur les "Waveforms"

3.6.1 Résultats expérimentaux

Nous présentons dans la figure 3.9, la sélection de variables, pour un PMC (a), et pour un RBF (b). Dans cette figure, on peut voir qu'un maximum des performances en validation est atteint lorsque le nombre de neurones supprimés est de 16. Cela correspond aux résultats des performances donnés dans le tableau 3.5. Les résultats montrent un gain considérable des performances, même en comparaison avec les autres techniques. On peut noter que la séparation des variables bruitées est bien plus importante lors de l'utilisation d'un RBF.

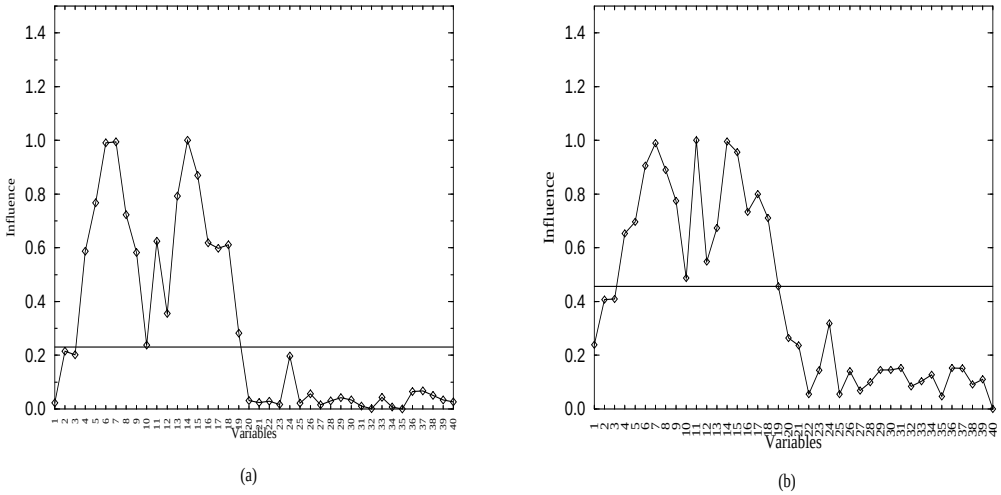

FIG. 3.9: *Influence en fonction des variables sur un MLP (a) et un RBf (b) (Les valeurs ont été réarrangées entre [0,1]). Les neurones ont été "prunés" par ordre d'influence (la plus basse en premier). La ligne horizontale donne les variables selectionnées lorsque les meilleurs performances ont été atteintes.*

	PMC avec 10 neurones cachés: <V | 10 | 3 >			RBF avec 39 références: <V | 39 | 3 >		
data set	D^l	D^v	D^t	D^l	D^v	D^t
40 variables V=40	92% ± 3.63%	83.86% ± 3.16%	82.84% ± 1.16%	93% ±3.46%	81.71% ± 3.29%	81.30% ± 1.19%
Var. sélect. V=16	88.67% ± 4.09%	85.57% ± 3.04%	85.37% ± 1.09%	88.33% ± 4.13%	86.86% ± 2.94%	85.60% ± 1.08%

TAB. 3.5: *Résultats des performances des deux architectures testées.*

3.6.2 Conclusion

Nous avons présenté dans cette section une nouvelle mesure permettant de déterminer l'influence d'un neurone sur un autre et ceci indépendamment de l'architecture utilisée. L'expérimentation montre de bonnes performances lors de l'utilisation de cette technique.
Cette technique peut elle aussi, être utilisée pour des problèmes d'optimisation d'architecture ainsi que pour l'extraction de règles. Ces deux problèmes font partie des perspectives à court terme que nous comptons développer.

3.7 Comparaisons

Nous présentons ici, une comparaison des différentes techniques de sélection de variable les plus connues, afin de valider IIIE et AINS. Cette comparaison, a été effectuée sur les vagues de Breiman, problème traité dans les validations précédentes.
Le tableau 3.6, montre que nos deux mesures IIIE et AINS, font partie des meilleurs sélecteurs, cela est d'autant plus vrai pour AINS appliqué au RBF. Il faut noté que pour HVS, il est possible d'atteindre les mêmes résultats que pour AINS, pour cela il suffit d'optimiser l'architecture à l'aide d'HVS.

	D^l	D^v	D^t	Variables sélectionnées
Toutes variables V=40	92.00% ± 3.63%	83.35% ± 2.90%	82.32% ± 1.15%	11111111111111111111 11111111111111111
Signal réel V=21	86.66% ± 4.31%	87.57% ± 2.65%	85.27% ± 1.08%	11111111111111111111 00000000000000000
OCD V=21	88.66% ± 4.08%	82.57% ± 2.99%	83.14% ± 1.14%	00011111111111111000 00100000100101011 00
HVS V=16	88.66% ± 4.08%	85.57% ± 2.80%	85.37% ± 1.09%	00011111111111111100 00000000000000000
IIIE V=16	88.66% ± 4.08%	85.57% ± 2.80%	85.37% ± 1.09%	00011111111111111100 00000000000000000
AINS PMC V=16	88.67% ± 4.09%	85.57% ± 2.80%	85.37% ± 1.09%	00011111111111111100 00000000000000000
AINS RBF V=16	88.33% ± 4.13%	86.86% ± 2.94%	**85.60%** ± 1.08%	00011111111111111100 00000000000000000

TAB. 3.6: *Performance pour le problème de Breiman.*
Les "1" correspondent au fait que la variable est sélectionnée, et les "0" au fait qu'elle ne l'est pas.

3.8 Conclusion

Nous avons présenté deux nouvelles mesures. Celles-ci ont été validées sur différents problèmes et comparées à d'autres techniques de sélections. Nous avons montré les possibilités de nos deux mesures. Elles ont l'avantage de donner des résultats comparables aux meilleures techniques.

Chapitre 3. Bibliographie

[Bennani 1998] Bennani (Y.). – Contributions au Contrôle de la Capacité de Généralisation des Systèmes d'Apprentissage Connexionnistes. *Thèse d'Habilitation à Diriger des Recherches à l'université de Paris-Nord*, 1998.

[Bollivier et al. 1991] Bollivier (M. De), Gallinari (P.) et Thiria (S.). – Cooperation of Neural Nets and Task Decomposition. *IJCNN'91*, vol. 2, 1991, pp. 573–576.

[Breiman et al. 1984] Breiman (L.), Freidman (J.), Olshen (R.) et Stone (C.). – Classification and Regression Trees. *Wadsworth Int. Group*, 1984.

[Cibas et al. 1994] Cibas (T.), Fogelman Soulie (F.), Gallinari (P.) et Raudys (S.). – Variable Selection with Optimal Cell Damage. *ICANN'94*, 1994.

[De bruin et al. 1988] De Bruin (A.), Rinnooy Kan (A.H.G.) et Trienekens (H.W.J.M.). – A simulation Tool for the performance Evaluation of Parallel Branch and Bound Algorithms. *Mathematical Programming*, vol. 42, 1988, pp. 245–271.

[Derijver et Kittler 1982] Derijver (P.A.) et Kittler (J.). – Pattern Recognition: a statitical approach. *Prentice-Hall International, London*, 1982.

[Fukunaga 1990] Fukunaga (K.). – Statistical Pattern Recognition. *Academic Press*, vol. 2, 1990.

[Hashem 1992] Hashem (S.). – Sensitivity Analysis for Feedforwrd Artificial Neural Networks with Differentiable Activation Functions. *International Joint Conference on Neural Networks, IJCNN'92*, vol. 1, 1992, pp. 419–424.

[Le cun et al. 1990] Le Cun (Y.), Denker (J.S.) et Solla (S.A.). – Optimal Brain Damage. *Neural Information Processing Systems*, vol. 2, 1990, pp. 598–605.

[Leray 1998] Leray (P.). – Apprentissage et Diagnostic de Systèmes Complexes: Réseaux de Neurones et Réseaux Bayésiens Application à la gestion en temps

réel du trafic téléphonique français. *Thèse d'informatique de l'université paris 6 (LIP6)*, 1998.

[Moody et Utans 1992] MOODY (J.) et UTANS (J.). – Principed Architecture Selection for Neural Networks: Application to Corporate Bond Rating Prediction. *Neural Information Processing Systems*, vol. 4, 1992.

[Moody 1994] MOODY (J.). – Prediction Risk and Architecture Selection for Neural Networks. *Statistics to Neural Networks-Theory and Pattern Rocgnition Application, Eds. V. Cherkassky, J.H. Friedmann, H.Wechsler, Springer-Verlag*, 1994.

[Ruck et al. 1990] RUCK (D.W.), ROGERS (S.K.) et KABRISKY (M.). – Feature selection using a multylayer perceptron. *Neural Network Comput*, vol. 2(2), 1990, pp. 40–48.

[Tresp et al. 1997] TRESP (V.), NEUNEIER (R.) et ZIMMERMANN (G.). – Early Brain Damge. *Neural Information Processing Systems*, vol. 9, 1997, pp. 669–675.

[Yacoub et Bennani 1997] YACOUB (M.) et BENNANI (Y.). – HVS : A heuristic for variable selection in multilayer artificial neural network classifier. *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 7, 1997, pp. 527–532.

Chapitre 4

Extraction de Règles


ègle, un petit mot
pour pouvoir tout régir
et pouvant faire si souffrir
en provoquant tant de maux.

L'extraction de règles est un problème complexe. Si la sélection de variables permet de déterminer les differentes variables importantes pour un problème, elle ne permet généralement pas d'exhiber les éléments importants pour chaque exemple. Nous présentons ici, un système d'extraction de règles issue de notre mesure IIIE et nous validons notre approche sur differents problèmes du domaine de l'apprentissage symbolique.

4.1 Extraction de règles

D'après l'étude précédente, nous avons montré l'utilité de la sélection de variables sur les performances d'un système connexionniste.
On peut, à l'aide de cette même technique, extraire des règles symboliques (comme on trouve dans [Bochereau et Bourgine1990], [Brezellec et Soldano1993], [Denoeux], [Turner et Gedeon], [Shavlik et Towell1991], [Gallant1988], [Konte et al. 1990]).
En effet, la méthodologie employée avec IIIE permet non seulement de sélectionner les variables sur un ensemble d'exemples mais d'extraire les caractéristiques pour chaque exemple d'une base.
En conséquence, si on entraîne un réseau de neurones artificiels sur la base d'un problème du domaine de l'apprentissage symbolique (comme pour l'apprentissage numérique cette base n'implique pas nécessairement que nous travaillons dans un univers clos), nous pouvons espérer obtenir les règles de décision implicitement codées dans ce réseau.

4.1.1 Protocole

Les problèmes du domaine de l'apprentissage symbolique ont comme particularité d'utiliser exclusivement les valeurs de vérité $vrai$ et $faux$. Nous avons, par conséquent, codé ces deux valeurs respectives par 1 et -1.
Les neurones d'entrée ont été associés à des variables propositionnelles, que nous noterons P_i. Chaque exemple du problème peut se voir comme une instanciation de ces variables propositionnelles que l'on associe à une valeur de vérité $vrai$ ou $faux$. C'est pourquoi tous nos réseaux auront la particularité d'avoir deux sorties représentant ces valeurs que nous noterons Q' et $\neg Q'$.
Un exemple sera donc de la forme :
$p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow Q$

Pour déterminer quelles sont les composantes pertinantes d'une règle, nous procèdons de la façon suivante:
On calcul pour un exemple la pertinence des variables à l'aide de IIIE, et nous concervons les variables dont la pertinence est supérieure à un seuil θ (celui-ci est calculé de façon à ce qu'il soit inférieure à la moyenne des pertinences de cet exemple). Notons que nous utilisons ici des poucentages que nous avons présentés dans le chapitre précédent, nous avons ainsi lorsqu'un pourcentage est negatif, l'information qu'il faut instancier la variable à $vrai$ si pour l'exemple la valeur est à $faux$ et à $faux$ dans le cas contraire.

Il faut noter que la méthode IIIE extrait une règle par exemple. Il nous a donc fallu concevoir une méthode de suppression et simplification des règles obtenues (Cf. Fig 4.1).

4.1.2 Méthode de suppression et simplification

Notations :

- R^i : règle extraite pour l'exemple i.
- R : ensemble des règles.
- P^{ni} : ensemble des variables propositionnelles sélectionnées pour l'exemple i;
- P^i : ensemble des variables propositionnelles pour l'exemple i;
- Q^{ni} : sortie du réseau pour l'exemple i.
- E_i : Exemple i de la forme $p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow Q$.
- B^i : Nombre d'exemples bien classés par la règle i.
- M^i : Nombre d'exemples mal classés par la règle i.
- Ens^i : Ensemble des exemples bien classés par la règle i.

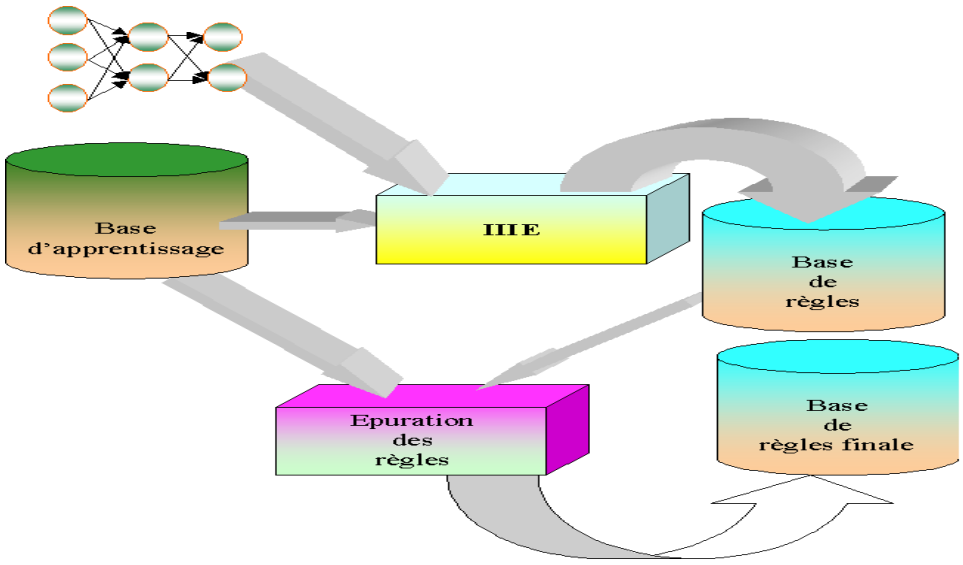

FIG. 4.1: *Système d'extraction de règles.*

Étapes de suppressions et de simplifications:

- Étape 1 : Suppression des règles redondantes.
  $\exists i, j \ R_j = R_i$ alors $R = R - \{R_i\}$

- Étape 2 : Association
  Associer pour chaque règles trois paramètres B^j, M^j, Ens^j que nous définissons comme suit :
  Si pour un exemple E^i, $\exists j$ $P^i \rightarrow P'^j$ et $Q'^j = Q^i$ alors $B^j = B^j + 1$ et $Ens^j = Ens^j \cup \{E^i\}$.
  Si pour un exemple E^i, $\exists j$ $P^i \rightarrow P'^j$ et $Q'^j \neq Q^i$ alors $M^j = M^j + 1$.

- Étape 3 : Suppression des règles dites mauvaises
  Si, $\frac{M^i}{B^i} < 0.1$ alors $R = R - \{R^i\}$.

- Étape 4 : Suppression des règles trop restrictives
  Si, $\exists j, i$ $Ens^j \subseteq Ens^i$ et $M^i \leq M^j$ alors $R = R - \{R^i\}$.

- Étape 5 : Favorisation des règles ayant le moins de prémisses
  Si $\exists j, i$ $Ens^j = Ens^i$ et $M^i = M^j$ et $\mid P'^i \mid > \mid P'^j \mid$ alors $R = R - \{R^i\}$.

Ces étapes permettent d'obtenir un ensemble de règles performantes et de taille réduite.

Complexité de cet algorithme:

Soit p le nombre d'entrées du réseau, n le nombre d'exemples du problème. On à alors:

- Complexité de l'étape 1:
  Sachant qu'une règle a au plus p composantes notés C_i, on peut coder chaque composante C_i sur 2 bits à savoir:
  00 : La composante C_i n'a pas été sélectionnée.
  01 : La composante C_i est sélectionnée est doit être instanciée à $vrai$.
  10 : La composante C_i est sélectionnée est doit être instanciée à $faux$.
  Ceci permet de coder une règle sur $2p$ bits, que l'on peut transformer aisément en entier.
  La recherche d'une règle revient alors à la recherche d'un entier parmis n entiers, en utilisant un simple algorithme de recherche par dichotomie la compexité est $log_2(n)$. Du fait que l'ensemble des règles se construit incrémentalement, l'étape a pour complexité: $log_2(\frac{n(n-1)}{2})$.

- Complexité de l'étape 2:
  Il suffit de regarder pour chaque règle si l'exemple est bien classé ou mal classé, la complexité est donc: n^2.
  Remarque: l'étape 3 peut être faite en même temps que l'étape 2 et ainsi n'augmente pas la complexité.

- Complexité de l'etape 4: Si l'on considère les exemples numérotés de 1 à n et qu'un exemple bien classés par une règle soit représenté par un 1 dans un

tableau de taille n, ou par 0 s'il est mal classé. La recherche de l'inclusion entre deux ensembles se fait avec une complexité de n.
La fait que l'ensemble des règles se construit incrémentalement, nous donne une complexité pour l'étape 4: $\frac{n^2(n-1)}{2}$.

- Complexité de l'étape 5:
  Cette étape peut se faire pendant l'étape 4 et nous n'avons pas d'ajout de complexité.

La complexité totale de l'algorithme est:

$$log_2(\frac{n(n-1)}{2}) + n^2 + \frac{n^2(n-1)}{2}$$

4.2 Validation

4.2.1 Problème du *ou-exclusif*

Ce problème est le même que celui traité pour la sélection de variables (Cf Fig. 4.2).

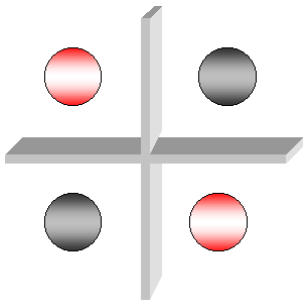

FIG. 4.2: *Problème du ou-exclusif, on observe bien que le réseau est confronté à un problème non linéairement séparable.*

L'énoncé diffère dans le sens où nous ne cherchons plus à déterminer les variables importantes, mais à connaître les différentes variables et instanciations adéquates pour obtenir une sortie correcte. La formalisation du problème est donc la suivante: on dispose de quatre variables propositionnelles P_1, P_2, P_3, P_4 et Q définie comme $P_2 \oplus P_4$.

La base d'apprentissage (cf tableau 4.1) correspond à la table de vérité du *ou-exclusif*.

Le réseau que nous avons utilisé (Cf Fig. 4.3) est de type PMC, avec 4 entrées et 2 neurones cachés et 2 neurones de sortie.

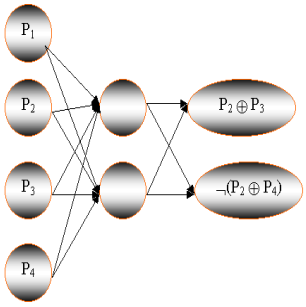

FIG. 4.3: *PMC pour le ou-exclusif.*

P_1	P_2	P_3	P_4	$P_2 \oplus P_4$
F	F	F	F	F
F	F	F	V	V
F	F	V	F	F
F	F	V	V	V
...	...	...	...	...
V	V	F	F	V
V	V	F	V	F
V	V	V	F	V
V	V	V	V	F

TAB. 4.1: *Table de vérité partielle du ou-exclusif.*

Le tableau 4.2 montre l'influence associée à chaque instanciation des variables propositionnelles pour chaque exemple. On montre, de ce fait, que le réseau a parfaitement associé les variables à la décision. Après l'élagage de la base de règles, on obtient un système (cf tableau 4.3) de règles permettant de déduire avec un taux de bonne classification de 100%, le problème du *ou-exclusif*.

I_1	P_1	I_2	P_2	I_3	P_3	I_4	P_4	règles obtenues
0%	F	48%	V	0%	F	49%	F	$P_2 \wedge \neg P_4$
0%	F	49%	F	0%	F	49%	V	$\neg P_2 \wedge P_4$
0%	F	49%	V	0%	V	49%	F	$P_2 \wedge \neg P_4$
0%	F	49%	F	0%	V	49%	V	$\neg P_2 \wedge P_4$
0%	V	49%	V	0%	F	49%	F	$P_2 \wedge \neg P_4$
0%	V	49%	F	0%	F	49%	V	$\neg P_2 \wedge P_4$
0%	V	49%	V	0%	V	49%	F	$P_2 \wedge \neg P_4$
0%	V	49%	F	0%	V	49%	V	$\neg P_2 \wedge P_4$

TAB. 4.2: *Influence (décimales tronquées) par variable et règle pour le ou-exclusif* .

$P_2 \wedge \neg P_4$
$\neg P_2 \wedge P_4$

TAB. 4.3: *Système de règles pour le problème ou-exclusif.*

4.2.2 Le problème $((P_2 \rightarrow P_1) \wedge (P_1 \rightarrow P_3)) \stackrel{?}{\rightarrow} (P_2 \rightarrow P_3)$

Pour ce problème l'idée est de vérifier qu'un réseau entrainé sur les variables propositionnelles P_1, P_2, P_3 avec Q définit par $(P_2 \rightarrow P_1) \wedge (P_1 \rightarrow P_3)$ permet d'en déduire l'implication $P_2 \rightarrow P_3$. Pour cela nous avons comme pour le problème du *ou-exclusif*, utilisé la table de vérité de Q (cf tableau 4.4) comme base d'apprentissage. Le réseau est aussi de type PMC, avec 3 neurones en entrée, 2 en couche cachée et deux en sortie (Cf Fig. 4.4).

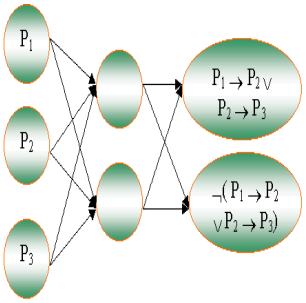

FIG. 4.4: *PMC pour le problème* $P_2 \rightarrow P_3$.

P_1	P_2	P_3	$(P_2 \rightarrow P_1) \wedge (P_1 \rightarrow P_3)$
F	F	F	V
F	F	V	V
F	V	F	F
F	V	V	F
V	F	F	F
V	F	V	V
V	V	F	F
V	V	V	V

TAB. 4.4: *Table de vérité pour le problème du* $(P_2 \rightarrow P_3)$.

On trouve dans le tableau 4.5 que le réseau a déterminé l'implication recherchée. Nous avons déterminé un système à trois règles (cf tableau 4.6).

I_1	P_1	I_2	P_2	I_3	P_3	règles obtenues
-4%	F	44%	F	-50%	F	$\neg P_1 \wedge \neg\neg P_3 \leftrightarrow \neg P_1 \wedge P_3$
37%	F	52%	F	10%	V	$\neg P_2 \wedge \neg P_1$
20%	F	42%	V	36%	F	$\neg(P_2 \wedge \neg P_3) \leftrightarrow P_2 \rightarrow P_3$
27%	F	42%	V	30%	V	Ne nécessite pas d'extraction
38%	V	29%	F	31%	F	Ne nécessite pas d'extraction
37%	V	9%	F	52%	V	$P_1 \wedge P_3$
19%	V	35%	V	45%	F	$\neg(P_2 \wedge \neg P_3) \leftrightarrow P_2 \rightarrow P_3$
3%	V	-55%	F	40%	V	$\neg P_2 \wedge P_3$

TAB. 4.5: *Influence par variable et règle pour le problème du* $(P_2 \rightarrow P_3)$.

Il faut noter que si un pourcentage est de valeur négative, cela implique qu'il faudra inverser sa valeur de vérité.

$\neg P_2 \wedge P_3$
$P_1 \wedge P_3$
$P_2 \rightarrow P_3$

TAB. 4.6: *Système de règles pour le* $(P_2 \rightarrow P_3)$.

4.2.3 Problème du *n parmi m*

Ce problème classique dans le monde de l'apprentissage symbolique (cf [Brezellec et Soldano1993]), consiste à déterminer parmi m variables propositionnelles, la combinaison permettant de déduire la valeur souhaitée.
Dans l'exemple que nous traitons, il s'agit de déterminer parmi 7 variables propositionnelles $P_1, P_2, P_3, P_4, P_5, P_6, P_7$ la combinaison pour que Q soit vraie si elle est définie comme $(P_i \wedge P_{i+1} \wedge P_{i+2})$ (Cf Fig. 4.5).

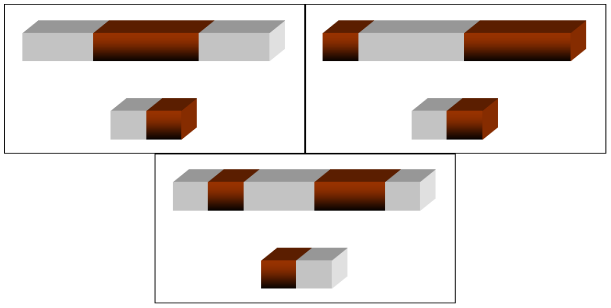

FIG. 4.5: *Représentation graphique de n parmis m.*

La base d'apprentissage est la aussi la table de vérité de Q (cf tableau 4.7). le réseau de type PMC employé est composé de 7 neurones d'entrée, 2 cachés et 2 neurones de sortie (Cf Fig. 4.6).

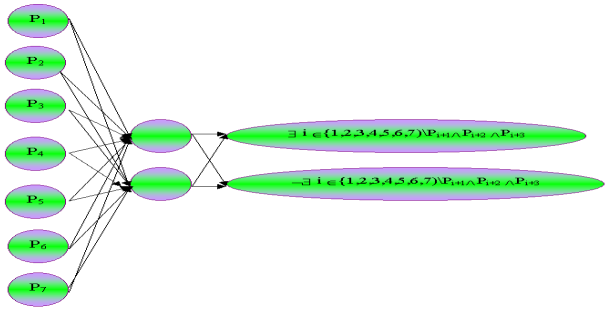

FIG. 4.6: *PMC pour le problème de n parmis m.*

P_1	P_2	P_3	P_4	P_5	P_6	P_7	$\exists i\ P_i \wedge P_{i+1} \wedge P_{i+2}$
F	F	F	F	F	F	F	F
F	V	V	V	F	F	F	V
V	F	F	V	V	V	V	V
V	V	V	V	V	V	V	V

TAB. 4.7: *Table de vérité partielle pour le problème du 3 parmi 7.*

Ce dernier problème confirme les premières expériences, à savoir la possibilité d'utilisation d'un réseau connexionniste pour l'extraction de bases de règles symboliques. Le tableau 4.8 montre l'émergence des règles induites de l'apprentissage permettant d'obtenir le système de règles énoncé dans 4.9.

I_1	P_1	I_2	P_2	I_3	P_3	I_4	P_4
0%	F	-19%	F	-21%	F	-1%	F
...	...	...	...	...	...	...	...
-14%	F	21%	V	3%	V	23%	V
...	...	...	...	...	...	...	...
7%	V	7%	F	10%	F	17%	V
...	...	...	...	...	...	...	...
5%	V	13%	V	21%	V	14%	V

I_5	P_5	I_6	P_6	I_7	P_7	règles obtenues
21%	V	21%	V	13%	V	$P_5 \wedge P_6 \wedge P_7$
...	...	...	...	...	...	...
-9%	F	-18%	F	-9%	F	$P_2 \wedge P_3 \wedge P_4$
...	...	...	...	...	...	...
26%	V	18%	V	11%	V	$P_4 \wedge P_5 \wedge P_6$
...	...	...	...	...	...	...
21%	V	15%	V	9%	V	$P_1 \wedge \cdots \wedge P_6$

TAB. 4.8: *Influence par variable et règle pour le problème de n parmi m.*

$P_1 \wedge P_2 \wedge P_3$
$P_2 \wedge P_3 \wedge P_4$
$P_3 \wedge P_4 \wedge P_5$
$P_4 \wedge P_5 \wedge P_6$
$P_5 \wedge P_6 \wedge P_7$

TAB. 4.9: *Système de règles pour le n parmi m .*

4.3 conclusion

Nous avons montré le potentiel d'un système connexionniste dans l'élaboration de règles symboliques. Ces travaux on été suivis par des expérimentations sur un problème réel relatif aux ARNs. Nous avons obtenu des résultats équivalents aux meilleures techniques symboliques[Bossaert1993].

Chapitre 4. Bibliographie

[Bochereau et Bourgine 1990] Bochereau (L.) et Bourgine (P.). – Extraction of semantic features and logical rules from a multilayer neural network. *IJCNN*, vol. 2, 1990, pp. 579–582.

[Bossaert 1993] Bossaert (F.). – Rapport de D.E.A. *Université paris XIII, Laboratoire d'Informatique de Paris-Nord (LIPN)*, 1993.

[Brezellec et Soldano 1993] Brézellec (P.) et Soldano (H.). – ELENA : A Bottom-Up learning method. *ICNL'93*, 1993, pp. 9–16.

[Cibas et al. 1994] Cibas (T.), Fogelman Soulie (F.), Gallinari (P.) et Raudys (S.). – Variable Selection with Optimal Cell Damage. *ICANN'94*, 1994.

[Denoeux] Denoeux (T.). – Génération automaitique de règles de classification par l'algorithme de rétropropagation du gradient. *AFCET AFIA, $3^{ème}$ journées du colloque symbolique numérique, Univ Paris IX.*

[Gallant 1988] Gallant (S.I.). – Connectionist expert systems. *ACM'88*, vol. 31, 1988, pp. 152–169.

[Konte et al. 1990] Konté (A.), Victorri (B.) et Raysz (J.P.). – Rule extraction in recurrent connectionist networks. *Neuro-Nimes'90*, 1990, pp. 131–144.

[Shavlik et Towell 1991] Shavlik (J.W.) et Towell (G.G.). – The extraction of refined rules from knowledge-based neural networks. *Machine Learning'91*, 1991.

[Turner et Gedeon] Turner (H.S.) et Gedeon (T.D.). – Extracting meaning from neural networks. $11^{ème}$ *journées d'intelligence artificielle d'Avignon.*

[Yacoub et Bennani 1997] Yacoub (M.) et Bennani (Y.). – HVS : A heuristic for variable selection in multilayer artificial neural network classifier. *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 7, 1997, pp. 527–532.

Chapitre 5

Gestion en Temps Réel du Trafic Téléphonique

Le Temps est invention,
ou il n'est rien du tout.

H. Bergson (1859-1941).

La gestion en temps réel du trafic téléphonique est un souci majeur dans le domaine de la téléphonie, en terme de qualité de service, surtout lors d'incidents pouvant survenir sur le réseau. Dans ce chapitre nous présentons dans un premier temps, le réseau téléphonique français. Nous introduisons ensuite les différents indicateurs du trafic qu'il nous a été possible d'utiliser, ainsi que le simulateur du trafic que nous avons eu à notre disposition. Enfin nous présentons de manière générale, les différentes méthodologies envisagées, pour répondre aux besoins de gestion.

5.1 Le réseau téléphonique Français

5.1.1 Architecture du réseau

La structure du réseau téléphonique commuté (RTC) Français est hiérarchique à quatre niveaux utilisant une politique d'acheminement fixe.
Cette structure est en quatre niveaux qui comprenait en 1993:

- 5 centres de transit principal (CTP),
- 80 centres de transit secondaire (CTS),
- 1000 centres à autonomie d'acheminement (CAA),
- 10000 centres locaux (CL),
- 30 millions d'abonnés.

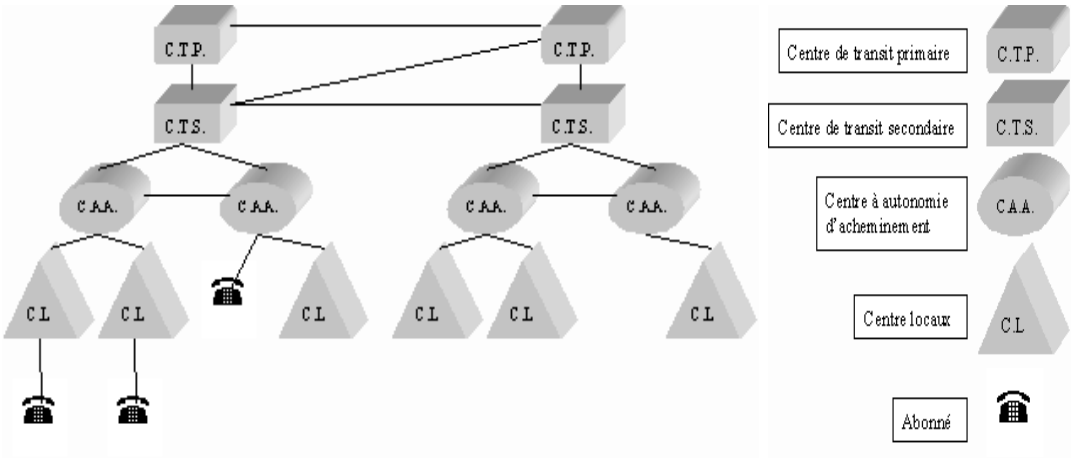

FIG. 5.1: *Structure hièrarchique du réseau téléphonique.*

5.1.2 Principales entités

Les principales entités constituant le réseau téléphonique sont les suivants:

- les centres : CTP, CTS, CAA, CL.
- Les faisceaux qui représentent un moyen logique pour modéliser les liaisons physiques (circuits) entre deux commutateurs.
  Il existe trois types de faisceaux:
  - les faisceaux directs :
    ils relient les centres de même niveau.

  - Les faisceaux hiérarchiques :
    ils relient les centres de niveau i aux centres de niveau $i+1$.
  - Les faisceaux transversaux :
    idêm que les faisceaux hiérarchique, mais les centres ne sont pas sur la même branche.

- Les flux de trafic qui représentent une demande en trafic existant entre un noeud d'origine et un noeud destination.

- Les différents acheminements qui recouvrent deux notions:

  - la politique d'acheminement utilisée dans le réseau,

  - la "Table" d'acheminement de chaque flux; les tables sont toutes regroupées en une seule.

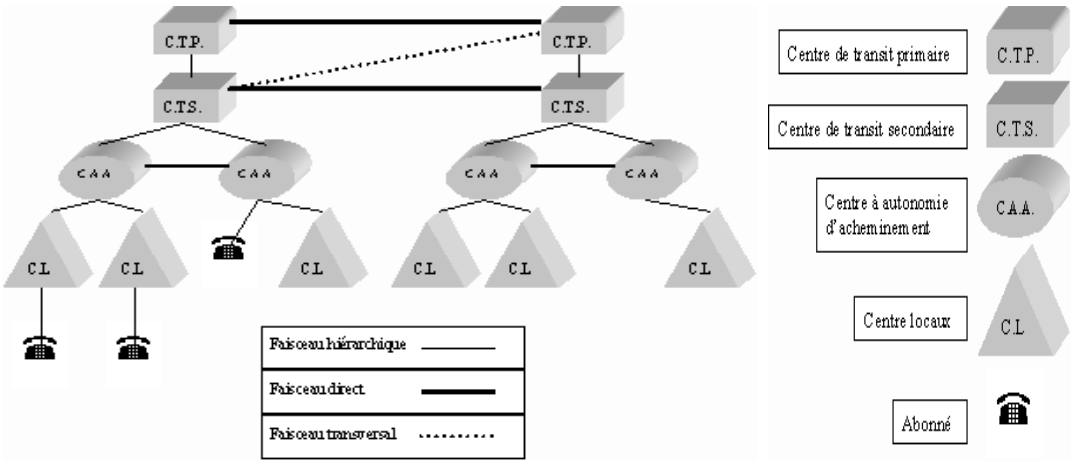

FIG. 5.2: *Structure hièrarchique des faisceaux.*

5.2 Diagnostic du réseau téléphonique

5.2.1 Perturbations du réseau

Pour notre étude, nous disposons de données réelles représentant le trafic des communications sur le réseau téléphonique français. Ces données sont issues d'un logiciel de France Télécom et a pour tâche de récuperer les valeurs provenant de différents capteurs de trafics disséminés sur le réseau. Ces mesures prises toutes les demi-heures pour minimiser la tailles des informations, ne font que donner une caricature de la fluctuation du trafic. En effet dans les centres de supervision du trafic téléphonique, les opérateurs reçoivent différentes mesures agrégées de la totalité du réseau, ces mesures ayant une périodicité variant de la seconde à la minute, de plus le fait que ces données soient recuillies directement sur le réseau

implique que l'on soit confronté a des capteurs défaillants et entraine ainsi une abscence de données. En outre ces informations représentants l'état du réseau ne sont pas étiquetées, ils ne peuvent nous donner une idée générale du trafic. C'est pourquoi, pour refléter le mieux les informations permettant aux opérateurs de faire du diagnostic sur le réseau, nous avons utilisé un simulateur afin de retrouver les périodicités qu'ont à leur disposition ces opérateurs et nous permettre d'étiqueter les différents états du réseau.

En fait, le réseau téléphonique constitue un système complexe au même titre qu'une machine de production, par exemple. C'est pourquoi la conception d'un système de diagnostic pour le superviseur représente un outil d'aide à la gestion. La tâche de diagnostic peut être décomposée comme une suite de module effectuant une tache spécifique à savoir un module de détection permettant de déterminer si un centre est dans un état perturbé, si tel est le cas le module identification a pour but d'identifier qu'elle est le type de perturbation que subit ce centre.

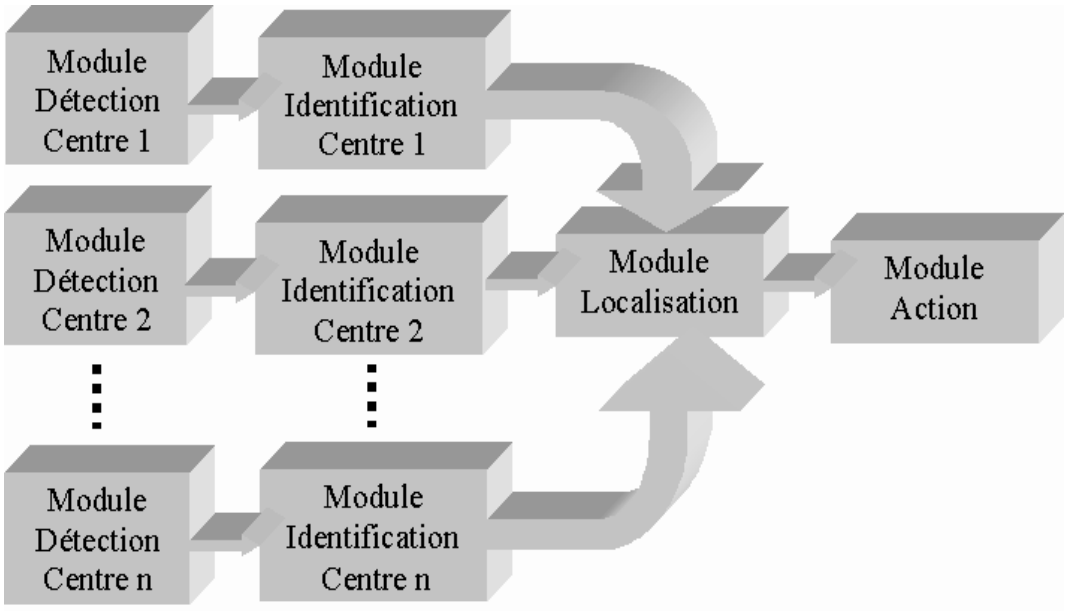

FIG. 5.3: *Schéma d'un système de diagnostic.*

Dans cette étude, nous avons abordé le problème de la détection et de l'identification de perturbations selon deux approches différentes :

- l'approche par modélisation
- l'approche par discrimination.

Nous allons brièvement décrire le simulateur dont nous nous servons dans cette étude ainsi que les problèmes de diagnostic que nous allons traiter.

La gestion du trafic est assurée dans les centres de supervision dans lesquelles le diagnostic et le contrôle s'appuient fortement sur des opérateurs. Les mesures qui peuvent être le nombre d'appels présentés a un centre (CTS ou CTP) ayant aboutis à une communication, ou bien encore le nombre d'appels présentés au centre n'ayant pas aboutis, etc.., sont regroupées et utilisées afin d'analyser l'état des centres correspondants, de détecter toute situation anormale (surcharge de trafic, incidents sur le réseau) et d'activer des commandes afin de minimiser les effets de la perturbation. Les différentes situations que nous devons reconnaître pour chaque centre sont les suivantes :

- SN - Situation nominale : tout est normal,
- SO - Surcharge origine : surcharge sur les flux partant d'un centre,
- SD - Surcharge destination : surcharge sur les flux arrivant à un centre,
- SG - Surcharge globale : augmentation de trafic sur tout le réseau,
- SR - Surcharge régionale : augmentation du trafic dans la zone d'un CTP.
- JT - Jeu téléphonique

Pour un CTS, les situations SG et SR sont très ressemblantes. Nous avons travaillé sur les diagnostics locaux, en considérant les variables d'états du trafic téléphonique correspondant au CTS (ou au CTP) afin de "diagnostiquer" le type de situation dans laquelle est ce centre, on pourra trouver une présentation plus fine dans [Didelet1992].

5.2.2 Diagnostic

La gestion en temps réel d'un réseau téléphonique est très complexe à réaliser. Différentes techniques ont été élaborées pour ce problème. On peut citer par exemple: l'analyse en composantes principales [Stern1991], la méthode du suivi [Raczkiewicz et Stern1993], l'utilisation des systèmes experts [Stern et Chemouil1992], les arbres de neurones [Didelet1994], les techniques classiques d'analyse des séries temporelles [De bois1994], les techniques de reconnaissance floue des formes [Boutleux et Dubuisson1995]. L'un des principaux problèmes de la gestion du trafic téléphonique est la détection en temps réel.

En effet, le fait que cette gestion à comme contrainte le temps réel, il s'avère qu'il n'est pas possible de concevoir un système qui consommerait trop de temps de calculs. C'est pourquoi nous nous sommes imposés la ligne de conduite suivante : élaborer des systèmes peu chers en temps de calculs, si bien que nous avons decomposé la tâche d'identification en deux composantes distinctes, à savoir la détection et l'identification. Cette dernière n'étant utilisée que dans l'éventualité où la détection aurait mis en évidence une anomalie.

La tâche de détection consiste, à partir d'indicateurs représentant l'état du réseau, à distinguer l'état nominal d'un état surchargé (Cf Fig. 5.3).

5.2.2.1 Détection

Nous avons étudié deux modèles de détections, ayant chacun des avantages et des inconvenients, nous présentons suscinctement dans ce paragraphe ces deux systèmes (Cf Fig. 5.4).

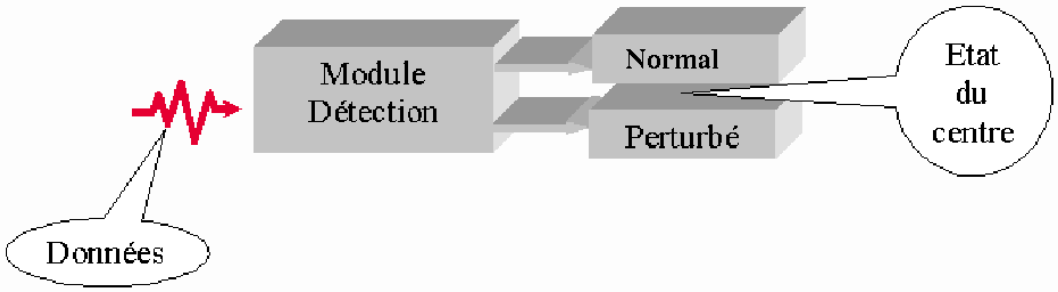

FIG. 5.4: *Schéma de détectection.*

Le premier système est un modèle discriminant (Cf Fig. 5.5). Le principe est simple, on associe à chaque observation du réseau son état nominal ou non, notre modèle apprendra à distinguer ces deux différent états. Le problème d'un tel système c'est que pour un bon fonctionnement, il faut connaître tous les états possibles du réseau.
De ce fait, si un nouvel état du réseau devait apparaître, le système deviendrait obsolète et toute la phase d'apprentissage serait à retravailler.

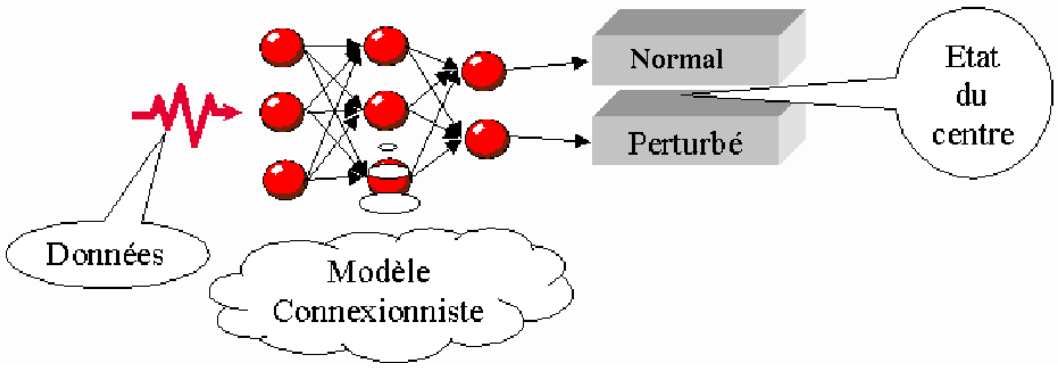

FIG. 5.5: *Schéma d'identification par une approche discriminante.*

Notre second système est basé sur les modèles prédictifs (Cf Fig. 5.6). Le principe est le suivant :

on élabore un modèle de l'état nominal du réseau. C'est à dire qu'en se basant sur un ensemble d'observations du passé, on établit un modèle qui apprendra à reproduire les valeurs que l'on devrait observer si le réseau restait dans cet état. Une fois ce modéle conçu, on examine la divergeance entre cette prédiction avec les obervations réelles. On détermine ainsi à l'aide d'un seuil de rejet l'état du réseau.
L'avantage de ce type de système est qu'il permet de gérer un nouvel état du réseau. En effet, il ne dépend que du modèle de l'état nominal. Cependant, il s'avère très délicat de déterminer le seuil de rejet. Nous developperons les différentes méthodes dans la suite de ce mémoire.

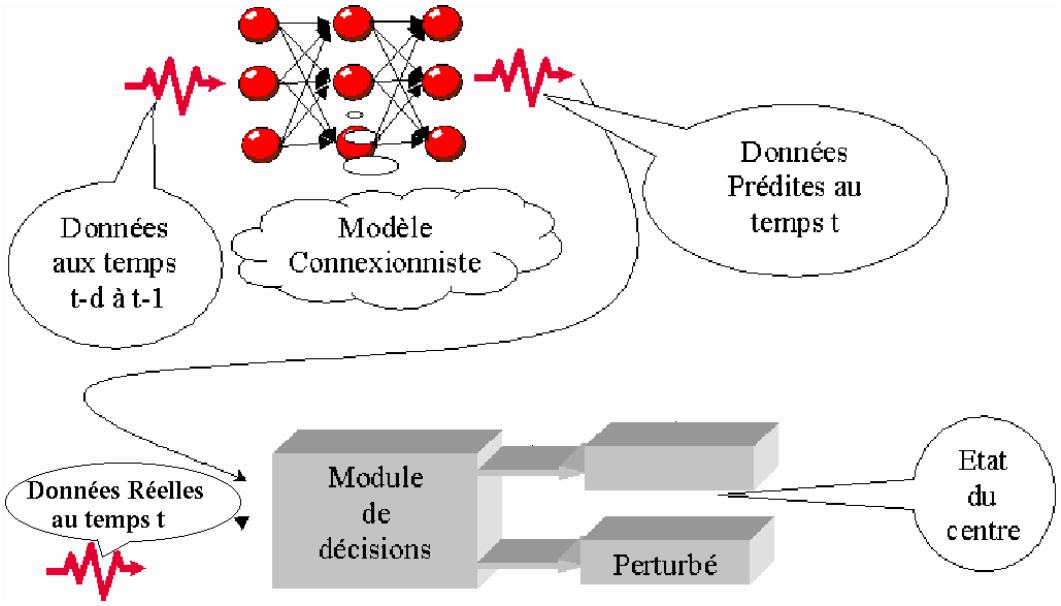

FIG. 5.6: *Schéma d'identification par modélisation.*

5.2.2.2 Identification

Ici nous présentons brièvement les deux approches que nous avons utilisées pour l'identification (Cf Fig. 5.7). Cette tâche pouvant être soit directement utilisée sans tâche de détections ou après avoir préalablement détecté une anomalie. Il est cependant préférable d'utiliser une phase de détection, car ce type de système est souvent plus coûteux en terme de temps calculs.

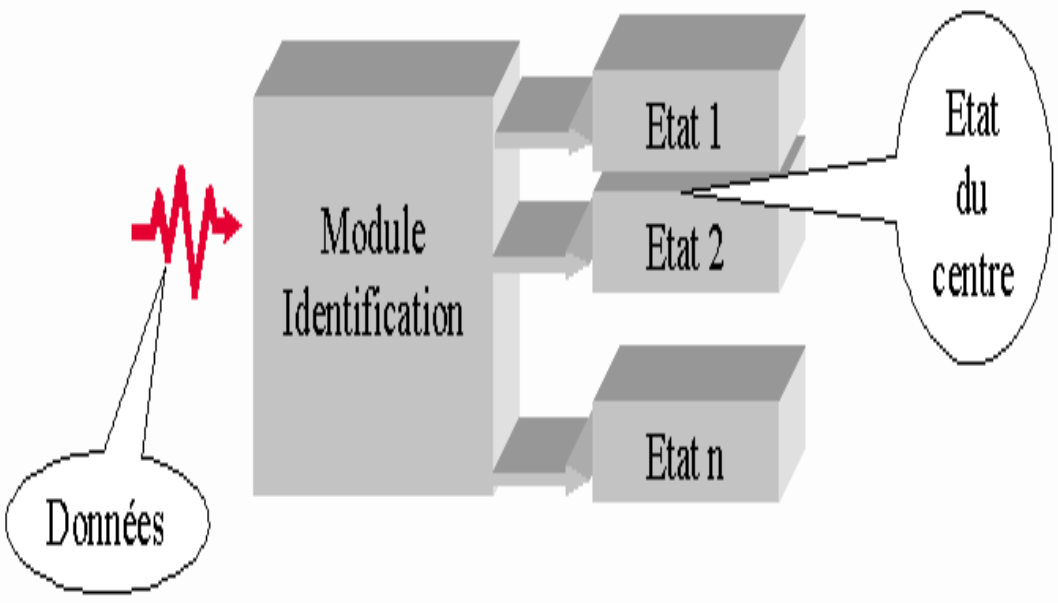

FIG. 5.7: *Schéma d'identification.*

Le système par approche discriminante est très similaire au modèle de détection, la différence entre ces deux systèmes réside essentiellement par la distinction entre les états perturbés. C'est à dire qu'au lieu d'associer les observations à un état nominal ou non, on associe chaque observation à l'état correspondant du réseau (Cf Fig. 5.8).
Ce système étant plus complexe, ses besoins en temps de calculs sont donc plus importants que le système de détection, qui reste de ce fait tout à fait naturel de conserver, afin d'utiliser l'identification à bon escient.

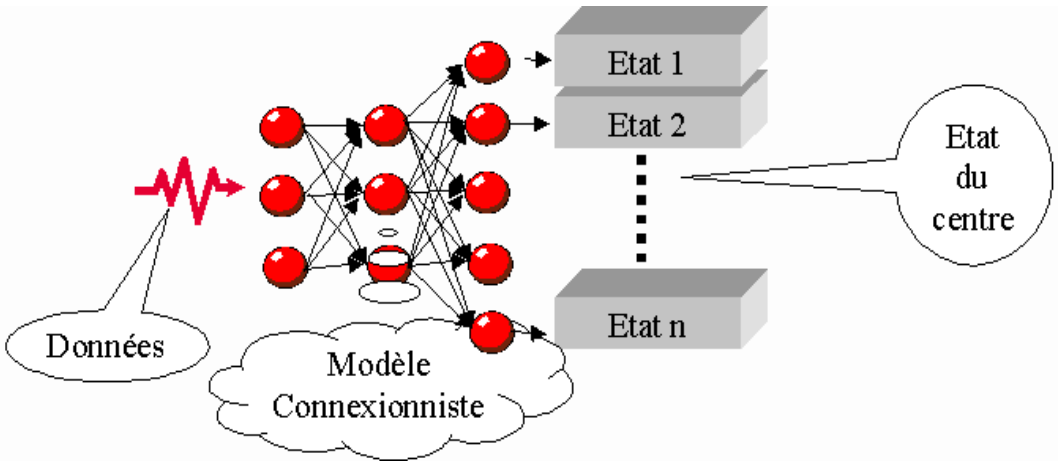

FIG. 5.8: *Schéma d'identification par modèle discriminant.*

Notre système par approche prédictive peut être vu comme un ensemble de système de détection (Cf Fig. 5.9), en fait on modélise ici chaque état du réseau. Le principe est basé sur la mise en concurence des différents modéles prédictifs, si bien que celui qui predira le "mieux" la valeur observée, entrainera l'association de l'état pour lequel il a été conçu à l'etat courant du réseau. L'idée est donc dans un premier temps de modéliser chaque état du système à diagnostiquer, dans un second temps le module d'identification met en compétition les différents modèles et les compares avec la situation réel du réseau. Cela permet en déterminant le modéle refletant le mieux la situation réelle observée de déterminer l'état du réseau.

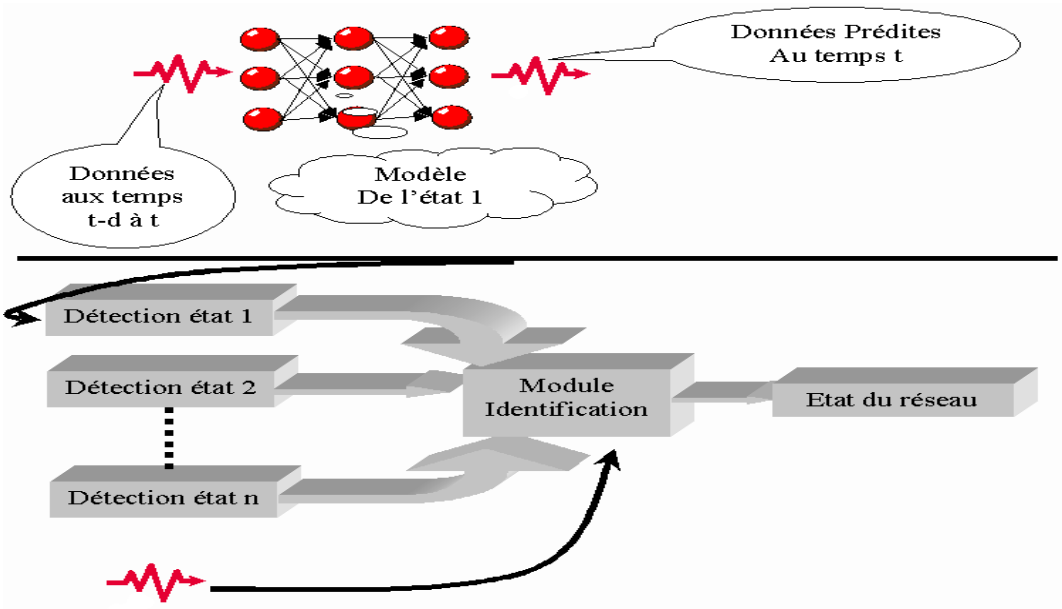

FIG. 5.9: *Schéma d'identification par modélisation.*

5.3 Simulation du trafic

5.3.1 Supermac

L'accès aux données réelles est possible par le logiciel Violette qui fourni un ensemble de mesures du trafic. Toutefois, réaliser une telle étude uniquement sur des données réelles pose plusieurs problèmes : accès aux données, données manquantes, représentativité des situations d'incidents (ces situations ne peuvent pas être générées à volonté sur le réseau réel). L'étude de systèmes complexes nécessite l'utilisation de simulateurs.
Le CNET a développé le simulateur SUPERMAC sur lequel nous nous sommes appuyés pour réaliser nos études. Bien qu'étant nécessaire, ce type de simulateur ne remplace cependant pas les données réelles. En effet, d'une part la conception

de scénarios représentatifs des phénomènes affectant le réseau peut devenir extrêmement complexe et nécessite une connaissance approfondie de l'évolution des indicateurs.
D'autre part certains phénomènes ne peuvent pas être modélisés (e.g. bruit aléatoire). SUPERMAC permet d'étudier la gestion du trafic en temps réel, il est bâti autour d'un simulateur événementiel reproduisant le plus fidèlement possible la demande des usagers et l'écoulement du trafic correspondant, le comportement du réseau, l'observation et la commande du trafic [Stern1994]. Dans SUPERMAC des mesures des indicateurs de trafic sont prises à intervalles de temps réguliers, et sauvegardées dans trois tables:

- une table pour les mesures relatives aux centres.
- une table pour les mesures relatives aux faisceaux.
- une table pour les mesures relatives aux flux qui composent les faisceaux.

5.3.2 Indicateurs du Trafic

Nous nous sommes intéressés uniquement aux mesures relatives aux centres, notons que certaines d'entre elles sont des mesures moyennes sur l'activité des faisceaux. Pour chaque centre, sont enregistrés 25 indicateurs de mesure offerts par SuperMac. Parmi les 25 variables, 7 ont été supprimées car elles ne concernent pas les CTS. Les variables considérées sont les suivantes, la numérotation donnée sera reprise dans toute la suite du mémoire.

1	Appels offerts
2	Appels offerts origine
3	Appels offerts destination
4	Appels échec distant
5	Prises efficaces
6	Prises efficaces origine
7	Prises efficaces destination
8	Appels efficaces
9	Appels efficaces origine
10	Appels efficaces destination
11	Trafic écoulé
12	Trafic écoulé origine
13	Trafic écoulé destination
14	Trafic efficace
15	Trafic efficace origine
16	Trafic efficace destination
17	Taux d'occupation
18	Taux d'efficacité

Table 1 : Variables utilisées pour l'étude d'un C.T.S.

Les mesures des indicateurs sont prises à intervalles de temps réguliers de 4 mn. Nous avons considéré six situations, correspondant au régime nominal et à cinq régimes de perturbation. Tous les scénarios partent du régime nominal et simulent une perturbation particulière. Les mesures traduisent plusieurs phénomènes :

- le réseau atteint tout d'abord l'état nominal. On attend la 12^e mesure à savoir apr
  ès que la 30^e minutes ait été simulée pour être sûr d'être dans cet état (Cette attente a été établie en colaboration des experts du CNET).

- la perturbation est ensuite déclenchée. On observe alors une montée en charge qui est assez brève dans le cas des scénarios étudiés.

- on atteint ensuite un régime stationnaire pour cette perturbation.

On considère qu'on est dans l'état surcharge dès le début de la montée en charge. Le simulateur utilise des nombres aléatoires pour simuler les appels téléphoniques. Il est possible d'agir sur cette suite de nombres pour obtenir différentes séries d'observations pour chaque situation de surcharge. Pour assurer la validité des résultats, nous avons ainsi généré plusieurs séries de mesures représentatives de chaque type de perturbation.
Les scénarios sont généralement de 2 heures. Les mesures étant prises toutes les 4 mn, le nombre de mesures par fichier est de 30. Seulement 19 mesures sont retenues car comme mentionné ci- dessus, les 11 premières correspondent à la montée en charge du réseau pour atteindre l'état nominal.
Chacun des six ensembles de données générées a été décomposé dans les proportions de 2/3, 1/3 en une base d'apprentissage et une base de test. Le tableau ci dessous donne la répartition du nombre d'observations pour chacune des situations.

	SN	SD	SO	SG	SR	JT	Total
Nb total de mesures	200	190	190	190	190	175	1135
Base d'apprentissage	133	130	130	130	130	115	768
Base de test	67	60	60	60	60	60	367

Table 2: constitution des bases d'apprentissage et de test

A partir de ces bases, nous avons étudié la faisabilité de la détection en se basant sur des méthodes simples, à savoir l'analyse en composante principale nous permettant de nous donner une vague idée de la confusion des données représentant différents états d'un centre (Cf. chapitre 9).

5.4 Conclusion

Dans ce chapitre, nous avons montré les différentes techniques qu'il est possible de mettre en oeuvre pour résoudre ce problème. Nous avons décrit les différentes données, et problèmes que nous devions aborder. Ceci nous a permis de mettre en évidence les difficultés qu'il fallait envisager et de confronter nos points de vue avec ceux des experts du CNET, afin de répondre au mieux à leurs besoins.

Chapitre 5. Bibliographie

[Boutleux et Dubuisson 1995] BOUTLEUX (E.) et DUBUISSON (B.). – Détection et Suivi d'Évolutions de l'État d'un Système Complexe: Application au réseau Téléphonique Français. *Personal Communication*, 1995.

[De bois 1994] DE BOIS (L.). – Time Series Forcasting or Network Traffic Management. *European Cooperation on Network Traffic Management*, 1994.

[Didelet 1992] DIDELET (E.). – Les arbres de neurones avec rejet d'ambiguité. Application au diagnostic pour le pilotage en temps réel du réseau téléphonique français. *Thèse de doctorat, Université Technologique de Compiègne*, 1992.

[Didelet 1994] DIDELET (E.). – A Neural Technique Approach to Network Traffic Management. *ITC 14, Antibes, France*, 1994.

[Raczkiewicz et Stern 1993] RACZKIEWICZ (M.) et STERN (D.). – Methods to Detect Traffic Disturbances or Real-Time Network Management. *Technical Repport, DE/ATR/82-93*, 1993.

[Saporta 1978] SAPORTA (G.). – Théorie et Méthodes de la Statistique. *Éditions Technip*, 1978.

[Stern et Chemouil 1992] STERN (D.) et CHEMOUIL (P.). – A Diagnosis Expert System or Network Traffic Management. *Networks92, Kobe, Japan*, 1992.

[Stern 1991] STERN (D.). – A Statistical Study o Real-Time Telephone Traffic Variations or Network Management. *ITC Specialist Seminar, Krakw, Poland*, 1991.

[Stern 1994] STERN (D.). – Supermac V3, 08 1994. CNET DE/ATR/08/94.

Chapitre 6

Génération d'Alarmes dans un Réseau Téléphonique

"Oui" et "non" sont les mots les plus courts et les plus faciles à prononcer et ceux qui demandent le plus d'examen.

C.M. de Talleyrand-Périgord (1754-1838).

La détection en diagnostic est la première tâche à accomplir. Même s'il semble simple de différencier un état nominal, d'un état perturbé, on est confronté au problème de discrimination des différents états. Cette détection devant se faire en temps réel, il est nécessaire d'utiliser des modèles peu coûteux en terme de temps de calcul. Nous présentons dans ce chapitre une première phase de détection par un modèle univarié, où celle-ci se fait par intervalle de confiance. Dans un deuxième temps, nous proposons une détection à base de modèles multivariés et nous étendons cette détection en modélisant des situations perturbées. Cela nous a conduit à définir une mesure permettant d'augmenter le pouvoir discriminant des modèles par réduction de dimensions.

6.1 Modélisation connexionniste univariée

6.1.1 Principe

Soit la série temporelle S représentée par les n valeurs $\{ x_1, x_2, \cdots, x_n \}$. La prédiction consiste à déterminer les valeurs futures $\{ x_{n+1}, x_{n+2}, \cdots \}$ à partir des valeurs passées de la série. En fait, concevoir un modèle de prédiction revient à trouver une fonction Ψ définie par un ensemble de paramètres W tel que :

$$X_t = \Psi(W, P_{t-1}^{t-d}) + e_t \ \ \forall t > d$$

ou

$$x_t^T = x_t^P + e_t \ \ \forall t > d$$

où :

- d est l'ordre du modèle,
- W représente l'ensemble des paramètres pour la prédiction,
- x_t^T est la valeur réelle de la série temporelle à l'instant t,
- x_t^P est la valeur prédite par le modèle,
- e_t est l'erreur de prédiction,
- P_{t-1}^{t-d} est le contexte de prédiction (le passé).

Le problème de prédiction consiste donc à déterminer W qui minimise, par exemple, la fonction de coût suivante :

$$C = \sum_{k \in S} e_k^2$$

Dans notre étude sur la détection des situations anormales nous avons utilisé une autre fonction de coût inspirée de [Bishop1994],[Nix et Weigend1995]. Cette nouvelle fonction de coût permet de déterminer simultanément la valeur prédite de la série temporelle et sa variance conditionnelle.
En effet, l'estimation de la valeur et de sa variance permet de calculer les bornes de l'intervalle de confiance de la valeur prédite (l'intervalle de prévision).

Nous supposons que les erreurs de prédiction sont distribuées normalement autour de x^p. Dans ce cas, la fonction de densité peut être donnée par l'expression suivante :

$$P(x_k^T | P_{k-1}^{k-d}) = [2\pi\sigma_k^2]^{-\frac{1}{2}} e^{-\frac{[x_k^T - \Psi(W, P_{k-1}^{k-d})]^2}{2\sigma_k^2}}$$

et le log de la vraissemblance est donné par :

$$-ln(P(x_k^T|P_{k-1}^{k-d}) = \frac{1}{2}ln(2\pi\sigma_k^2) + \frac{[x_k^T - \Psi(W, P_{k-1}^{k-d})]^2}{2\sigma_k^2}$$

La sommation sur tous les exemples donne :

$$C(W) = \frac{1}{2}\sum_{k\in S}(\frac{[x_k^T - \Psi(W, P_{k-1}^{k-d})]^2}{\sigma_k^2} + ln(\sigma_k^2) + ln(2\pi))$$

La fonction de coût que nous avons choisie, permettant l'estimation simultanée des deux statistiques, est donnée par :

$$C(W) = \sum_{k\in S}\left\{\frac{[x_k^P - \Psi(W, P_{t-1}^{t-d})]^2}{\sigma_k^2} + \ln\sigma_k^2 + \ln 2\pi\right\}$$

Une idée similaire a été utilisée pour une tâche d'approximation [Thiria et al. 1992]. Elle consiste à faire des histogrammes sur les valeurs des sorties désirées. Pour cela, l'espace des sorties est discrétisé en un nombre fini d'intervalles. On transforme alors une sortie réelle en un vecteur binaire où chaque position représente l'appartenance de cette sortie à un intervalle. Les sorties calculées peuvent ensuite être interprétées comme des probabilités d'appartenances à l'intervalle. Cette approche modélise de façon discrète la distribution des sorties du réseau.

En utilisant la fonction de coût $C(W)$, l'apprentissage consiste donc à prédire la valeur à l'instant t et l'intervalle de prévision, c'est-à-dire l'intervalle dans lequel il y a une probabilité de $p\%$ d'avoir la valeur réelle observée.
Notre système (Cf Fig. 6.1) est basé sur la coopération de deux modules : le premier apprend à prédire la valeur à l'instant t (moyenne) et le second module apprend à prédire la variance de cette valeur. Les variables d'entrées ayant été déterminées par l'études de corrélogramme et des matrices de corrélations des indicateurs (Cf. Chap 9).

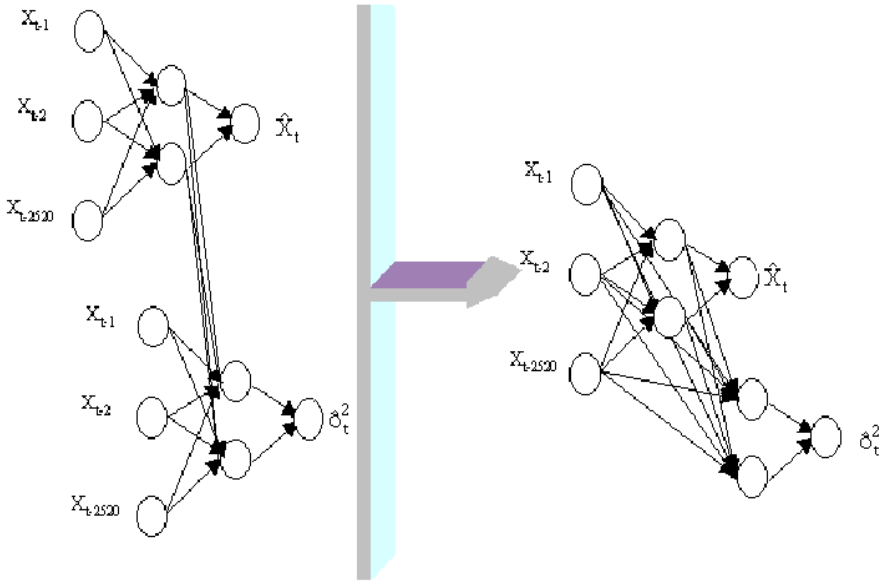

FIG. 6.1: *Architecture modulaire pour la prédiction.*

6.1.2 Protocole de l'apprentissage

La phase d'apprentissage comporte trois étapes:

1- apprentissage du premier module:
Consiste à estimer la moyenne conditionnelle en utilisant une base A pour l'apprentissage et une base B pour la validation.

2- aprrentissage du deuxième module:
concerne l'estimation de la variance conditionnelle en utilisant la base B pour l'apprentissage et la base A pour la validation.

3- apprentissage global:
comporte l'approximation des deux statistiques simultanément en utilisant cette fois une base C pour l'apprentissage et une base D pour la validation.

6.1.3 Principe de la détection

Les sorties de l'architecture modulaire: $\hat{X}_t$ et $\hat{\sigma}_t^2$, nous permettent de calculer l'intervalle de prévision dans lequel doit se trouver la valeur réelle observée avec une probabilité de $p\%$ ($P(z < k) = p\%$). $\Omega_t = [\hat{x}_t - k\sigma, \hat{x}_t + k\sigma]$ si cette valeur est dans l'intervalle alors il n'y a pas d'anomalie, sinon une anomalie est détectée:

$$\begin{cases} \text{Si } X_t \in \Omega_t \text{ Alors état normal} \\ \text{Si } X_t \notin \Omega_t \text{ Alors anomalie} \end{cases} \tag{6.1}$$

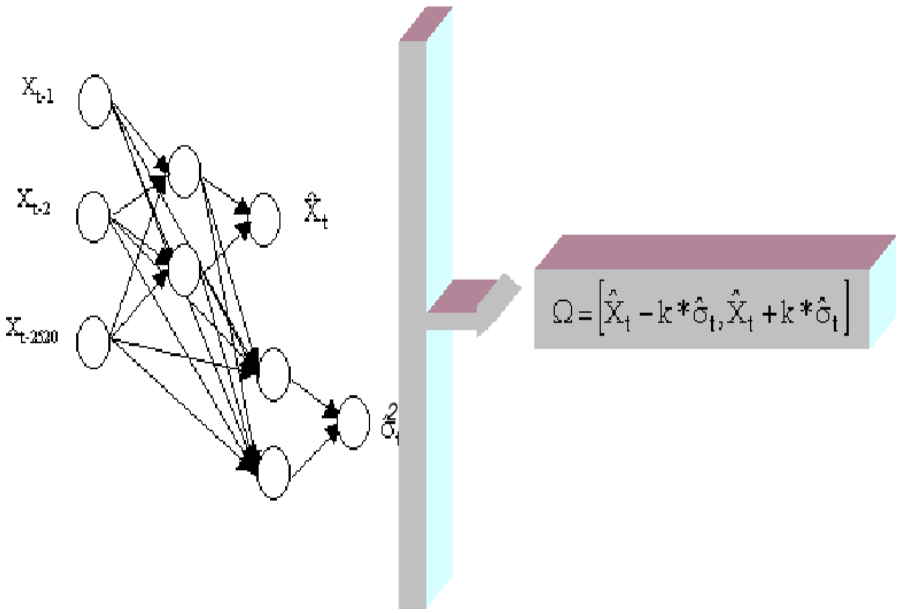

FIG. 6.2: *Principe de calcul de l'intervalle de prévision.*

6.1.4 Validation

Pour générer la base de données, nous avons utilisé SuperMac [Herrmann et al. 1989]. Nous nous sommes basés sur les études précédentes [Stern et Chemouil1992] du problème. Nous avons utilisé les trois indicateurs: Outgoing Bids (OB), Incoming Seizures (IS), Outgoing Seizures (OS) et les trois rapports: OS/IS, OB/IS et OS/OB. Pour modéliser l'état normal, nous avons simulé 8 semaines pour l'apprentissage et la validation. Ces 8 semaines sont réparties en 4 ensembles de 2 semaines chacun: A,B,C,D. Pour l'état anormal nous avons généré 4 semaines ayant des perturbations de différents types réparties en 2 ensembles E et F de 2 semaines chacun. Les instant de perturbations ont été choisis aléatoirement. Les différentes perturbations utilisées sont: Surcharge globale (SG), Surcharge destination (SD), Surcharge origine (SO) et Incident flux (IF). La période déchantillonage est de 4 minutes. Nous avons sur la figure 6.3 une sortie écran de notre système présenté lors d'une réunion avec le CNET. Cette figure est composée des points suivant:

- En haut de la figure, Une semaine de la base d'apprentissage pour un indicateur.
- En bas de la figure, la semaine de test utilisé.
- Au centre de la fenêtre, une journée perturbée représentée par la courbe noire, les deux courbes encadrant celle-ci correspondent aux bornes de l'intervalle de prévision fournie par notre système.

– En bas de la fenêtre, des traits verticaux correspondent aux différentes détections de notre système.

– A droite de la fenêtre, l'architecture modulaire neuronale utilisée pour notre système.

Cette expérience montre que les pics correspondant aux incidents provoqués sont en dehors de l'intervalle d'où une détection de notre système. Toutefois, nous pouvons remarquer à l'aide des traits qu'après certaines anomalies les données ne reviennent pas immédiatement à un état normal et qu'à certaines heures de la journée les perturbations provoquées sont imperceptibles par notre système (Incident flux, surcharge origine, ...).

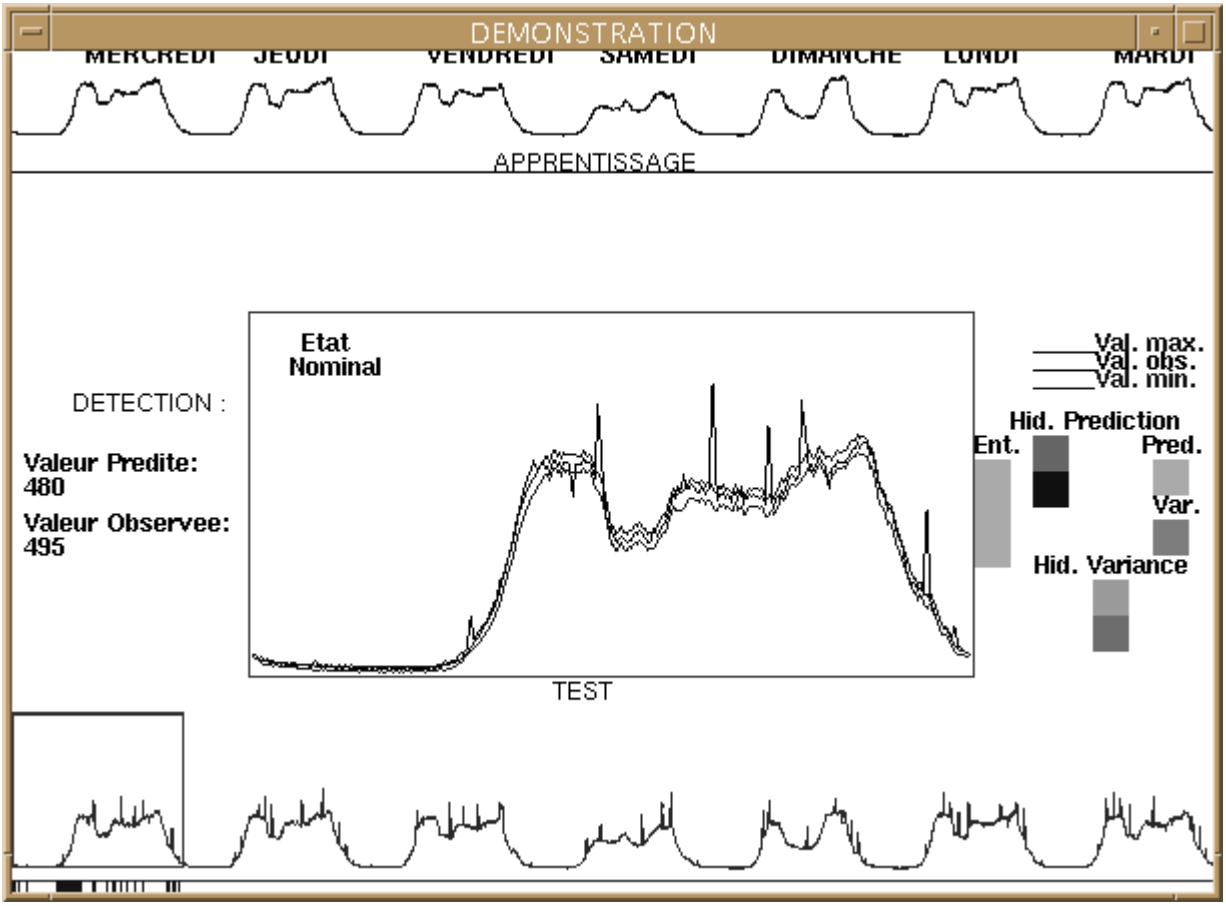

FIG. 6.3: *Interface du système de détection.*

6.1.5 Analyse et comparaisons des résultats de détection

Les résultats de la détection que nous avons analysés dans un premier temps en fonction de l'indicateur (ou rapport d'indicateurs) utilisé, nous ont permis de

déterminer un choix de trois statistiques émergeant de cette première étude afin d'obtenir une analyse plus fine de notre système, à savoir :

- BD : taux de bonne détection (le système génère une alarme en présence d'une anomalie)
- ND : taux de non détection (le système ne génère pas d'alarme en présence d'une anomalie et considère quil s'agit d'un état nominal)
- FD : taux de fausse détection (le système génère une alarme en présence d'un état normal)

Les taux BD et ND sont calculés sur le nombre d'anomalies présentes dans la base de test F (805 anomalies), et le taux FD est calculé sur le nombre de situations normales de la base F (4017 situations normales).

Notre analyse de ces résultats confirme les premières remarques que nous avions faites c'est-à-dire qu'après certaines anomalies les données ne reviennent pas immédiatement à un état normal et qu'à certaines heures de la journée les perturbations provoquées sont imperceptibles par notre système (Incident flux, surcharge origine, ...). On peut donc emettre l'hypothèse qu'un seul indicateur ne permet pas de déterminer totalement l'état du réseau.
Pour palier au problème de "débordement des incidents", nous avons décidé d'étiqueter les deux mesures qui suivent la fin de l'incident comme un état normal.
Le choix du système de référence permettant de comparer les performances de notre système connexionniste de façon empirique s'est porté sur la technique Follow-Up développé dans [Stern1991].
Celle-ci se base sur la moyenne ainsi que l'ecart type, son principe est le suivant :

- S_t : une série temporelle.
- x_t : la valeur de la série à l'intant t.
- μ_t^d : la moyenne des $x_i, i \in \{t-d, \cdots, t-1\}$.
- σ_t^d : l'ecart type des $x_i, i \in \{t-d, \cdots, t-1\}$.
- ν : correspond au coordonnée de la table de la loi normale, sachant que k est le degré de confiance, on a $P(z < \nu) = k$.
- d : la taille de la fenêtre.

L'idée de cette technique est de prédire la valeur de la série à l'instant t à partir des d valeurs précédentes de la série. On obtient une valeur prédictive $\widehat{x}_t$, sous

cette forme :

$$\widehat{x}_t = \mu_t^d + \nu \frac{\sigma_t^d}{\sqrt{d}}$$

c'est-à-dire que l'on considère que la valeur observée est conforme à la prédiction si et seulement si :

$$x_t \in [\mu_t^d - \nu \frac{\sigma_t^d}{\sqrt{d}}, \mu_t^d + \nu \frac{\sigma_t^d}{\sqrt{d}}]$$

Nous avons déterminé de façon empirique une taille de fenêtre adéquate pour notre système, en partant du choix fait pour la technique Follow-up. Notre choix s'est arrêté sur une taille de 10 mesures.

6.1.5.1 Analyse en fonction du type d'indicateur

Les résultats de détection pour nos différents indicateurs sont portés en tables 6.1 et 6.2. Si on s'intéresse par exemple à l'indicateur IS, on peut voir qu'il ne va pas de soit qu'un indicateur qui minimise FD (fausses détections) maximise les BD (bonnes détection).

De cette remarque, il ressort qu'il faut déterminer s'il est préférable de choisir un indicateur qui minimise ND ou un indicateur maximisant FD.
Après étude de ce phénomène et discussions avec des personnes du CNET, notre choix s'est porté sur des indicateurs (ou combinaisons d'indicateurs) qui offrent un bon compromis entre ND et FD. En fait choisir un indicateur qui minimise au mieux ND et FD. On peut prendre par exemple le rapport d'indicateurs OS/IS.

	OB		IS		OS	
	χ	ϕ	χ	ϕ	χ	ϕ
Bonnes Détections	324 40%	128 16%	278 35%	130 17%	469 58%	127 16%
Non Détections	481 60%	677 84%	527 65%	675 83%	336 42%	678 84%
Fausses Détections	679 17%	268 06%	490 12%	321 07%	1103 07%	299 22%

TAB. 6.1: *Résultats de détection en fonction du type d'indicateurs (χ : Système Connexionniste, ϕ : système de référence avec une fenêtre de 10, BD et ND sont calculés sur 805 anomalies, FD est calculé sur 4017 états nominaux, les résultats sont donnés sous forme de nombre d'événements et sous forme de pourcentage par rapport au nombre total)*

	OS/IS		OB/IS		OS/OB	
	χ	ϕ	χ	ϕ	χ	ϕ
Bonnes Détections	584 73%	167 21%	528 66%	165 21%	563 70%	064 08%
Non Détections	221 27%	638 79%	277 34%	640 79%	242 30%	741 92%
Fausses Détections	1335 18%	898 03%	1285 27%	739 33%	1698 32%	136 42%

TAB. 6.2: *Résultats de détection en fonction du type d'indicateurs (χ : Système Connexionniste, ϕ : système de référence avec une fenêtre de 10, BD et ND sont calculés sur 805 anomalies, FD est calculé sur 4017 états nominals, les résultats sont donnés sous forme de nombre d'événements et sous forme de pourcentage par rapport au nombre total)*

En comparant les performances du système de référence (Méthode Folow-up) avec les performances de notre système connexionniste, on peut clairement noter une nette supériorité de l'approche connexionniste.
Il est indubitable que si l'on porte son attention sur les taux de BD (Bonne détection) et ND (Non détection) notre système connexionniste offre les meilleurs résultats, cependant a contrario le taux de FD (Fausse détection) est moindre pour le système de référence, car nous avons une moyenne de 1098 fausses détections pour notre système contre 444 pour le système de référence et il est donc clair que celui-ci minimise bien mieux le nombre de fausse détection.
Cette différence de comportement des deux systèmes s'explique par le calcul de l'intervalle de confiance propre à chacun. Celui-ci est très pénalisant dans le système connexionniste, si bien qu'une partie de l'état nominal sera considérée comme anomalie, contrairement au système de référence qui a un large intervalle de confiance qui lui évitera ce problème, mais ne détectera pas de ce fait certaines anomalies.
Il faut noter que ces expériences ont été faites sans tenir compte d'éventuelles combinaisons d'indicateurs, et ainsi apporter certaines informations essentielles. Comme nous allons voir dans le paragraphe suivant.

6.1.5.2 Analyse en fonction du type d'anomalies

Choisir une bonne combinaison d'indicateur qui maximiserait BD et minimiserait les taux ND et FD, peut être déterminée en analysant en fonction du type d'anomalie la non détection pour chacun des indicateurs.
La répartition des taux de non détection en fonction du type d'anomalies associé à chacun des indicateurs est donnée en tables 6.3.
On peut résumer l'information contenue dans cette table en montrant pour chaque type d'anomalie le ou les indicateurs maximisant le mieux le taux de bonnes dé-

tections.

- Incident Flux, le meilleur rapport c'est avec OS/IS.

- Surcharge Origine et Surcharge Destination, c'est avec le rapport OB/IS.

- Surcharge Globale, les meilleurs rapports sont OB et OS.

Il faut remarquer que OB et OS sont très fortement corrélés.
La conception de notre système de détection basé sur cette analyse, recevra donc en entrée un ensemble d'indicateurs (ou de rapport d'indicateurs) ayant comme particularités de minimiser pour chacun ND un type d'incident donné. Ainsi, on pourra augmenter BD tout en diminuant ND.
Au niveau comparaison de notre système avec le système de référence, les remarques faites dans le paragraphe précédent restent vraies. De plus les résultats des tables 6.3 et 6.4 révèlent un comportement identique du système connexionniste et de référence (Follow-Up) que celui observé, lors de l'analyse des non détections, c'est-à-dire qu'il semble que la dètection à certainnes heures de la nuit soit très difficile.

Non Détections

	OB		IS		OS	
	χ	ϕ	χ	ϕ	χ	ϕ
IF	190 82%	213 93%	189 82%	207 90%	105 45%	214 93%
SO	076 38%	154 77%	155 77%	181 90%	071 35%	151 75%
SD	153 70%	195 89%	096 44%	172 78%	098 44%	172 78%
SG	062 39%	115 72%	087 55%	115 72%	062 39%	115 72%

TAB. 6.3: *Résultats de non détection en fonction du type d'incidents (χ : Système Connexionniste, ϕ : système de référence avec une fenêtre de 10, les résultats sont donnés sous forme de nombre d'événements et sous forme de pourcentage par rapport au nombre total du type d'anomalie)*

Non Détections

	OS/IS		OB/IS		OS/OB	
	χ	ϕ	χ	ϕ	χ	ϕ
IF	050 21%	188 82%	148 64%	182 79%	074 32%	216 94%
SO	035 17%	149 74%	012 06%	152 76%	062 31%	174 87%
SD	029 13%	180 82%	024 11%	185 84%	045 20%	205 94%
SG	107 67%	121 76%	093 58%	121 76%	061 38%	146 92%

TAB. 6.4: *Résultats de non détection en fonction du type d'incidents (χ : Système Connexionniste, ϕ : système de référence avec une fenêtre de 10, les résultats sont donnés sous forme de nombre d'événements et sous forme de pourcentage par rapport au nombre total du type d'anomalie)*

6.1.5.3 Analyse en fonction de la tranche horaire

Il s'avère que pour mieux comprendre le système de détection et ainsi mieux évaluer la difficulté du problème, l'analyse des résultats de la détection en fonction des tranches horaires est cruciale.
C'est pourquoi sur la figure 6.4, nous présentons l'histogramme de la répartition par tranche horaire des non détections par type d'incidents, extrait du comportement de notre système lors de l'utilisation de l'indicateur OB, celui-ci est intéressant du fait de son grand nombre de fausses détections, l'analyse des autres indicateurs n'étant pas présentée ici pour ne pas surcharger d'informations redondantes.
De ces résultats, il est à noter deux points :

- L'incident de type IF est peu ou pas détecté par le système et ceci est valable quelque soit l'indicateur utilisé, de plus cette non détection porte sur l'ensemble des tranches horaires observées.

- Les incidents qui se présentent pendant la première tranche horaire T1 (] 00h00 - 07h00]) ne sont généralement pas détectés.

Ces comportement peuvent être dus à deux raisons principales, soit à une mauvaise modélisation du comportement du processus durant cette tranche horaire, soit au fait que le taux de surcharge soit relatif au nombre d'appel présenté pendant ces heures, l'augmentation n'est donc pas forcement très significative.

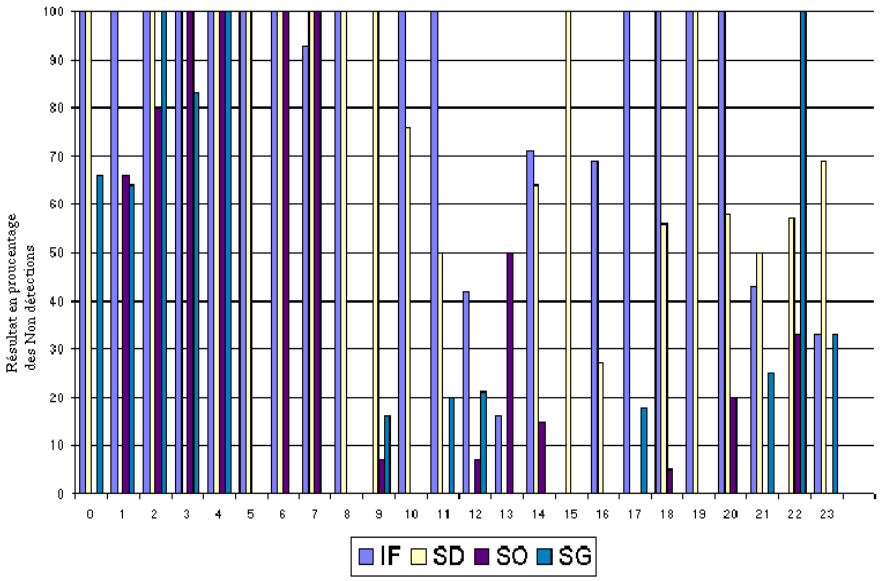

FIG. 6.4: *Résultats de non détection en fonction du type d'incidents et tranche horaire pour l'indicateur OB (Connexionniste)*

6.1.6 Conclusion

En conclusion, l'analyse des différents résultats de détection révèle que la qualité de la détection est fortement dépendante du choix de l'indicateur utilisé, ainsi qu'à la tranche horaire considérée.
Les performances de notre système connexionniste comparées aux performances du système de référence ont montré, un net avantage pour l'approche connexionniste. D'autant plus que les faibles résultats observés au niveau de la tranche horaire T1, sont vraisemblablement dus à un taux trop faible des surcharges, et ne sont donc pas essentiels pour le bon fonctionnement du réseau.

6.2 Modélisation Modulaire et Intervalle de Confiance

A partir des résultats de détection analysés dans les paragraphes précédents, nous avons établi le rôle majeur des indicateurs, ainsi que celui du choix des techniques connexionnistes dans le bon fonctionnement du système de détection. Pour améliorer les systèmes proposés, nous nous sommes portés sur des systèmes multi-modulaire. Pour ce faire, ce système comporte pour les entrées une combinaison d'indicateurs, issue des analyses de détections préalablement effectuées et de l'étude des matrices de corrélations.

6.2.1 Principe et critères de performance

Pour cette étude, nous avons utilisé les mêmes modèles connexionnistes que précédemment, à savoir un PMC et un RBF. Pour améliorer ces deux modèles nous avons changé les méthodes d'apprentissages que nous avions utilisées, en passant pour le PMC à l'algorithme du gradient conjugué pouvant permettre d'éviter certains minima locaux, et au second ordre pour l'architecture RBF afin de déterminer automatiquement le pas d'apprentissage; élément essentiel pour l'apprentissage de ce type d'architecture.

Pour résumer, nous disposons dans ce système multi-modulaire de 6 PMC, qui approximent chacun l'évolution d'un de nos 6 indicateurs (ou rapport d'indicateurs), montrés lors de nos études précédentes.
Dans notre système nous avons introduit un paramètre supplémentaire noté α intervenant comme coéficience pondérateur de l'écart type pour augmenter ou réduire l'intevalle de confiance, celui-ci a été déterminé par validation croisée afin de maximiser le taux des BD, minimiser le taux des FD, pour chaque tranche horaire et chaque architecture différente. Ceci en utilisant les critères de maximisation des bonnes détections (Critère C_1) et de minimisation des fausses détections (Critère C_2).
On peut trouver les résultats de ce travail, dans les tables 6.5 et 6.6, où l'on trouve respectivement les résultats pour la maximisation des BD et les résultats pour la minimisation des FD.
Nous avons pour ce système, introduit une légère modification dans le protocole précédemment utilisé, à savoir :
Après les détections du système, s'il s'agit réellement d'une situation perturbée, nous considérerons que la perturbation se termine lorsque notre système considérera être revenu en situation nominale. Ce nouveau protocole s'explique par le fait qu'une perturbation, ne se dissout pas immédiatement après l'avoir arrêté, mais reste perceptible pendant une durée qu'il n'est pas possible de déterminer a priori.

Dans la table 6.5, nous pouvons noter que le taux de bonnes détections est tout à fait satisfaisant, cependant reste le problème du taux des fausses détections qui demeure important. A contrario, au niveau de la table 6.6 on constate un faible taux des fausses détections, avec le problème symétrique, c'est-à-dire un taux de bonnes détections plus faibles.

6.2.2 Validation

Pour résoudre le problème des fausses détections, en gardant un taux satisfaisant de bonnes détections, nous sommes partis sur une combinaison de différentes

Résultat Global

	OB		IS		OS	
	PMC	RBF	PMC	RBF	PMC	RBF
BD	79%	71%	93%	98%	92%	60%
FD	39%	31%	67%	83%	65%	24%

Bonnes détections

	OB		IS		OS	
	PMC	RBF	PMC	RBF	PMC	RBF
IF	73%	41%	91%	94%	91%	33%
SO	85%	67%	88%	90%	94%	42%
SD	72%	49%	97%	97%	90%	38%
SG	86%	59%	96%	90%	95%	31%

TAB. 6.5: *Maximisation des bonnes détections : Critère* C_1

Résultat Global

	OB		IS		OS	
	PMC	RBF	PMC	RBF	PMC	RBF
BD	48%	44%	50%	62%	46%	50%
FD	04%	03%	02%	06%	02%	02%

Bonnes Détections

	OB		IS		OS	
	PMC	RBF	PMC	RBF	PMC	RBF
IF	27%	11%	36%	29%	32%	19%
SO	63%	62%	22%	31%	63%	55%
SD	22%	12%	65%	71%	10%	19%
SG	65%	59%	64%	59%	65%	54%

TAB. 6.6: *Minimisation des fausses détections : Critère* C_2

architectures qui permettent d'obtenir un faible taux de fausses détections. Nous avons donc établi un nouveau critère à maximiser, à savoir C_3 qui est basé sur la différence entre les bonnes détections et les fausses détections, critère déterminé en colaboration avec les experts du CNET. Ainsi, lors de sa maximisation, on peut espérer trouver un taux de bonnes et fausses détections satisfaisant.

Le principe de la détection en généralisation est le suivant: soit Ω_t^i l'intervalle de prédiction de l'architecture i a l'instant t,
s'il existe un j tel que x_t (valeur observée) n'appartient pas Ω_t^j
alors nous considérons que le système émet une alarme.
Comme précédemment énoncé, nous ne considérerons pas comme situation nominale les mesures suivantes d'une perturbation, si le système considère toujours être en situation d'anomalie.
Le fait que nous ayons 12 architectures différentes (un PMC par indicateur d'où 6 PMC et un RBF par indicateur d'où 6 RBF), a soulevé un problème combinatoire. Il nous a en effet fallu déterminer la meilleure combinaison parmi un ensemble de 4079 ($\sum_{2 \leq j \leq 12} C_{12}^j = 2^{12} - 1 - 12$) combinaisons possibles. C'est-à-dire que nous avons du trouver le meilleur système qui maximise le critère C_3 suscité. Ce système est présenté sur la figure 6.5, on peut voir sur cette figure un module de décision, il a pour rôle de déterminer s'il y a détection ou non, en éffectuant un vote majoritére des modules de détections.

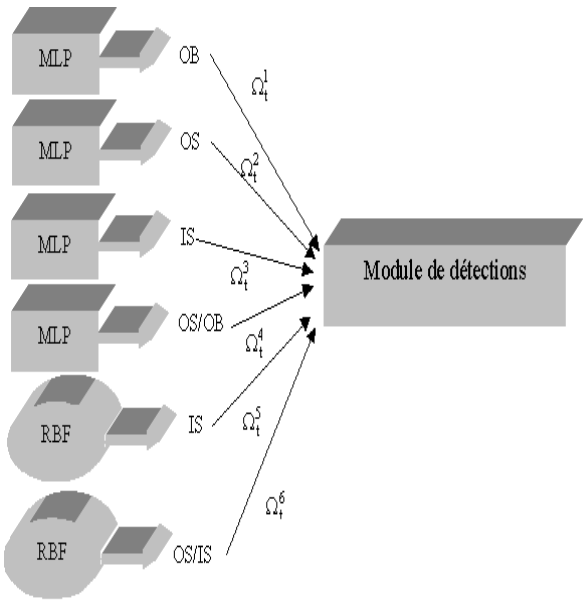

FIG. 6.5: *Meilleure combinaison suivant le critère Max BD-FD*

Dans la table 6.7, nous avons les résultats donnés par ce système (C_1) et

nous avons ajouté les résultats de la combinaison donnant le meilleur taux de bonnes détections (C_2). Sur la figure 6.6 nous présentons l'évolution des bonnes détections et des fausses détections, ainsi que l'évolution de la différence entre ces deux taux.

Résultat Global

	C_3	C_1
BD	77 %	84 %
FD	09 %	19 %

Bonnes Détections

	C_3	C_1
IF	54 %	68 %
SO	88 %	68 %
SD	83 %	93 %
SG	53 %	89 %

TAB. 6.7: *Taux de bonnes détections suivant le système choisi*

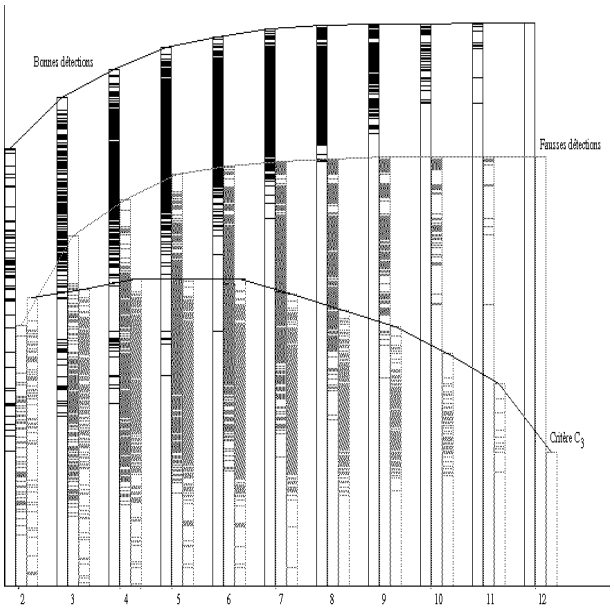

FIG. 6.6: *Évolution en fonction des combinaisons du taux des bonnes détections, des fausses détections et du critère C3.*

Nous avons montré qu'utiliser un système composé de différentes architectures (PMC et RBF) prédisant différents indicateurs permet d'avoir un taux correcte de bonnes détections, tout en gardant un taux de fausses détections acceptable. Cependant un certain nombre de points reste à expliquer. Dans un premier lieu, le fait qu'à certaines heures de la journée le système ne détecte pas les perturbations,

et que la perturbation de type incident flux reste très délicat à détecter.
Le phénomène qui se produit pour les heures creuses de la journée s'explique par le fait que nous avons utilisé des incidents ayant des taux de surcharges relatifs au nombre d'appels qui se présentent habituellement à cette heure.
Hors à ces heures de la journée; les communications sont très faibles et même si le taux de surcharge reste le même (en valeur relative), il devient très faible lorsque l'on s'intéresse aux valeurs numériques. Ce qui tant à rendre impossible la détection.
Au niveau des incidents flux, le phénomène reste semblablement équivalent au phénomène précédemment cité, c'est-à-dire que ce type d'incident porte sur un flux entrant vers le centre, et ce qui est peu important par rapport au nombre de flux entrant au centre, il n'est donc pas très perturbateur pour le trafic.

6.3 Modélisation Multivariée et Région de Confiance

6.3.1 Principe

Jusqu'à présent, les modélisations que nous avons effectuées ne portaient que sur une seule variable, en utilisant une architecture de type Nix-Weigend [Nix et Weigend1995]. A l'aide de ce modèle nous déterminions un intervalle de prédiction. Appliqué à la détection d'anomalie, ce système nous a permis de déterminer si une mesure observée se trouvait dans cet intervalle et donc de considérer que cette mesure était compatible avec le modèle de la série, afin de déterminer les mesures perturbées.
Nous sommes tenus à étendre ce principe au cas vectoriel, afin de gérer l'évolution simultanée de plusieurs variables, appliqué à notre problème en fait modéliser l'état nominal par un ensemble d'indicateur. Pour ce faire, nous avons élargi le concept d'intervalle de confiance pour le cas scalaire à un calcul de région de confiance. L'idée étant de se baser sur des méthodes de calculs de région de confiance pour des modèles prédictifs linéaires. Ici nous avons utilisé comme base la méthode de Bonferroni [Jobson1991].
Celle-ci permet à partir d'un modèle linéaire prédictif de définir une région de confiance pour les prédictions. Le principe du calcul est le suivant :
Soit le modèle prédictif linéaire :

$$\hat{X}_{t+1} = b_0 + b_1 X_t^1 + b_2 X_t^2 + \cdots + b_n X_t^n$$

- $\hat{X}_{t+1}$ est l'estimation de X_{t+1} par le modèle linéaire,
- b_i sont les paramètres du modèle linéaire,

dans le cas vectoriel Bonferroni définit une approximation de la région de confiance pour cette estimation , comme suit :

$$\hat{X}_{t+1} \pm coef * \sigma * (X_t'(X'X)^{-1}X_t)^{\frac{1}{2}}$$

avec

$$\begin{aligned} X_t &= (X_t^1, X_t^2, \cdots, X_t^n) \\ (X'X) &= \text{ matrice de corrélation des variables des différents } X_t \\ coef &= \text{ un facteur déterminé à partir de tables statistique.} \\ \sigma &= (\sigma_1, \sigma_2, \cdots, \sigma_n) \text{ estimation de l'écart type des variables } X_t^1, X_t^2, \cdots, X_t^n. \end{aligned}$$

6.3.2 Adaptation au cas des modèles connexionnistes

Dans le cas des réseaux connexionnistes prédictifs, nous avons utilisé la même méthode pour le calcul des régions de confiance. Les réseau que nous avons utilisé possèdent deux couches de poids : une première non linéaire et une deuxième linéaire. Cette dernière couche linéaire peut être vue comme un modèle linéaire prédictif. L'application de la méthode de Bonferroni sur cette couche est aisée et permet d'approximer des régions de confiance pour les prédictions des réseaux. La première couche effectue un codage qui est ensuite présenté à la seconde couche qui est un modèle linéaire.
L'approximation de la région de confiance peut être faite de la manière suivante :

$$\hat{X}_{t+1} \pm coef * \sigma * (h_t'(H'H)^{-1}h_t)^{\frac{1}{2}}$$

avec

$$\begin{aligned} h_t &= h_t^1, h_t^2, \cdots, h_t^n \text{ activations de la couche cachée} \\ &\quad \text{du réseau connexionniste prédictif (RCP)} \\ (H'H) &= \text{ matrice de corrélation des } h_t \text{ calculée sur la base d'apprentissage} \\ coef &= \text{un facteur déterminé par validation croisée,} \\ \sigma &= (\sigma_1, \sigma_2, \cdots, \sigma_n) \text{ estimation de l'écart type des variables, avec} \\ \sigma_i^2 &= \sum_i (S_i - d_i)^2 \end{aligned}$$

L'approximation de Bonferroni et l'utilisation de la dernière couche cachée des RCP comme modèle linéaire, nous permettent de calculer la région de prévision Ω dans laquelle doit se trouver la valeur réelle observée X_{t+1}. Cette région fournit un encadrement de la valeur approchée $\hat{X}_{t+1}$.
Une telle région encadrant la valeur $\hat{X}_{t+1}$ est donnée par :

$$\Omega = \left[\hat{X}_{t+1} - coef * \sigma * (h_t'(H'H)^{-1}h_t)^{\frac{1}{2}}; \hat{X}_{t+1} + coef * \sigma * (h_t'(H'H)^{-1}h_t)^{\frac{1}{2}}\right]$$

Si la valeur réelle observée X_{t+1} est dans la région de confiance alors il n'y a pas d'anomalie, sinon une anomalie est détectée :

$$\begin{cases} \text{Si } X_{t+1} \in \Omega & \text{alors état nominal} \\ \text{Si } X_{t+1} \notin \Omega & \text{alors état perturbé} \end{cases}$$

Ce système étant élaboré dans le but de différentier l'état normal du système à diagnostiquer d'un état perturbé sans distinction de la perturbation.

6.3.3 Validation

On présente dans les tables 6.8 et 6.9 les résultats de l'approche prédictive incluant la technique de calcul de région de confiance que l'on compare avec une approche basée sur la classification directe.
L'architecture du RPC utilisé a comme entrée les 18 indicateurs décrits dans le chapitre précedent représentant l'état du réseau à l'instant t et comme sortie ces mêmes indicateurs à l'instant $t+1$. L'architecture du réseau connexionniste à classification directe a comme entrée les 18 indicateurs à l'instant t et 2 sorties représentant l'état du réseau à l'instant $t+1$. Les architecture exactes sont données sous la forme $< in|hid|out >$, à savoir in correspond au nombre de neurones de la couche d'entrée, hid au nombre de neurones de la couche cachée et out au nombre de neurones de la couche de sortie.

Approche	Taux Moyen	Intervalle de Confiance à 95%*
Prédictive	84,88 %	[83,24 % , 86,38 %]
$< 18\|18\|18 >$		
Classification	91.67 %	[90.29 % , 92.66 %]
$< 18\|9\|2 >$		

TAB. 6.8: *Performances des deux approches en détection.*

$*$ L'intervalle de confiance est calculé par la formule $I_\alpha = \frac{P+\frac{Z_\alpha^2}{2n} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N}+\frac{Z_\alpha^2}{4n}}}{1+\frac{Z_\alpha^2}{n}}$
où n est le nombre d'exemples,
P est le taux de classifications
et $Z_\alpha = 1.96$ le degré de liberté dans le cas d'une probabilité $\alpha = 95\%$.
Il apparaît en étudiant la table 6.8 que l'approche par classification directe est plus intéressante (au niveau performance) par rapport au modèle prédictif à région de confiance. Cependant, si nous faisons une analyse plus fine, on remarque à partir des matrices de confusions que l'état nominal est bien mieux détecté avec l'approche par modélisation, et que l'état perturbé est sensiblement mieux identifié par l'approche classification directe.
Cette différence s'explique principalement par le nombre de données utilisées. Si l'on pointe l'ensemble d'apprentissage pour l'approche modélisation, on s'aperçoit que seules les données état nominal ont été utilisées; contrairement au modèle classification directe qui utilise l'ensemble des perturbations.

Approche Modélisation	Situations Nominales	Situations Perturbées
Situations Nominales	99.50%	0.50%
Situations Perturbées	18.70%	81.30%

Approche Classification	Situations	Situations
Situations Nominales	77.35 %	22.65 %
Situations Perturbées	4.76%	95.24%

TAB. 6.9: *Matrices de confusion des deux approches en détection*

6.3.4 Conclusion

La méthode d'approximation de Bonferroni et l'utilisation de la dernière couche cachée des RCP comme modèles linéaires, nous permettent de calculer la région de confiance dans le cas vectoriel où nous avons utilisé simultanément l'évolution de plusieurs indicateurs.

A la suite de cette étude nous avons entrepris d'étendre cette méthode à l'inditification.

6.3.5 Validation sur de nouvelles données

Pour cette étude, nous avons généré de nouvelles bases de données, à savoir 31 jours dont 200 mesures perturbées par jour, soit 11 160 mesures toutes bases confondues. Les bases ont été découpées comme suit :

- 15 jours pour l'apprentissage,
- 8 jours pour la validation,
- 8 jours pour le test.

Les figures 6.7 et 6.8 montrent l'évolution temporelle et spatiale des 18 indicateurs pour les 5 états étudiés du réseau : état nominal, et les 4 états perturbés.

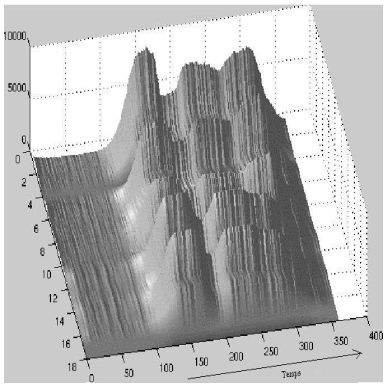

FIG. 6.7: *Évolution temporale et spatiale des indicateurs pour la situation nominale.*

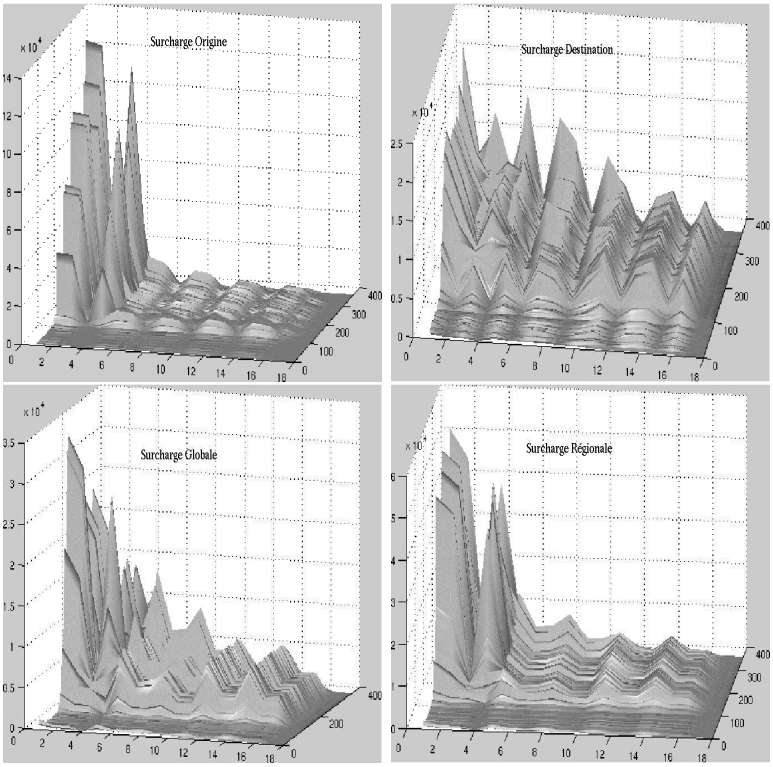

FIG. 6.8: *Évolution temporale et spatiale des indicateurs pour le 4 situations perturbées.*

Nous avons ainsi étudié la détection de ces 5 états du réseau à l'aide d'un RPC combinées au calcul de région de confiance. Pour comparer ces résultats nous avons appris un PMC en classification directe ayant la même tâche de détections que le RPC.

Le tableau (table 6.10) présente les résultats de validation sur la base de test de l'approche basée sur le calcul de région de confiance et l'approche de discrimination directe. Ces résultats montrent que les modèles de classifications directes et prédictifs donnent sensiblement les mêmes résultats, cependant le modèle de classification pose moins de problèmes en fausse détection.

Architecture	Discrimination 88,12% [87,39%;88,81%]	
$< 18\|9\|2 >$	BD	FD+ND
SN	70,61% 1129	29,39% 470
SD	95,56% 1528	04,44% 71
SO	95,81% 1532	04,19% 67
SG	86,49% 1383	13,51% 216
SR	92,12% 1473	07,88% 126

Architecture	Modélisation 87,93% [87,20%;88,63%]	
$< 18\|18\|18 >$	BD	FD+ND
SN	51,53% 824	48,47% 775
SD	96,00% 1535	04,00% 64
SO	99,62% 1593	00,38% 06
SG	95,75% 1531	04,25% 68
SR	96,75% 1547	03,25% 52

TAB. 6.10: *Classification Directe vs Modélisation & Régions de confiance*

6.3.6 Analyse de l'influence des variables sur la qualité de modélisation

Nous avons montré précédemment que de déterminer l'ensemble des variables réellement utile pour une classification, augmente significativement le taux de bonne performance (Cf chapitre 3). Cependant les techniques connues à ce jour même si elles peuvent être utilisées sur des modèles prédictifs ne permettent pas dans ce cas d'intégrer de discrimination.
En fait elles ne répondent pas au problème suivant :

- Est-ce qu'une variable facilement modélisable est bonne pour discriminer un modèle?

Cette question ne peut être résolue, en n'étudiant que le modèle prédictif; du fait qu'il n'intégre pas d'information discriminante sur les autres états. Il faut prendre en compte l'information inter-modèle en gérant tous les modèles prédictifs en même temps. Pour apporter une première solution au problème, nous avons utilisé un critère de mesure de qualité de prédiction ARV (Averaged Relative Variance) défini comme suit :

soit S une série temporelle (des indicateurs)

$$arv(S) = \frac{\sum_{t \in S} (x_t^o - x_t^p)^2}{\sum_{t \in S} (x_t^o - \mu)^2}$$

avec

x_t^o : valeur observée à l'instant t
x_t^p : valeur estimée par le réseau à l'instant t
μ : moyenne de la série S

Remarques
Si $x_t^p \approx \mu$ alors $arv(S) = 1$ et si $x_t^p = x_t^o$ alors la prédiction est exacte et $arv(S) = 0$.

Le modèle de prévision est d'autant meilleur que l'arv est proche de zéro. Notre idée est d'éliminer les variables qui perturbent la modélisation, afin d'augmenter la précision de notre région de confiance. Pour cela on supprimera les neurones ayant une arv grande pour l'état nominal et faible dans les autres états.

6.3.6.1 Procédure du calcul de l'influence d'une variable sur la prédiction

Sur les 5 bases de validation $(B_1, \cdots, B_5)$ correspondant aux 5 états possibles du réseau, on calcule la matrice ARV (représentée par le tableau 6.11) de chaque variable en utilisant le modèle de l'état nominal, on obtient ainsi une matrice 18x5.

ARV	B_1	B_2	B_3	B_4	B_5
V_1					
V_2					
V_i				$arv(V_i, B_4)$	
$\cdots$					
V_{18}					

TAB. 6.11: *Matrice ARV*

On définit le rapport :

$$I(V) = \frac{\frac{1}{N-1} \sum_{1<j\leq N} arv(V, B_j)}{arv(V, B_1)}$$

avec

B_j : série j de validation (représentant l'état j du réseau)
B_1 : série de l'état nominal
N : nombre d'états
V : variable
$arv(V, B_j)$: arv calculée pour la variable V en utilisant la série B_j.

Remarque :
le rapport I(V) a comme particularité d'être petit lorsque l'arv de l'état nominal est grand pour cette variable V (ie: la variable V "perturbe" la qualité de prédiction de l'état nominal) et les arv dans le cas des autres états sont petites (ie: la variable V participe à la qualité de prédiction pour les autres états).
Le principe d'élagage consiste à ne considérer que les variables (neurones de sortie) maximisant le rapport I(v), cela permet de conserver les variables d'un modèle prédisant mal les autres.

6.3.6.2 Résultats après élagage

L'application de ce principe donne les résultats suivant (Cf Fig. 6.9).

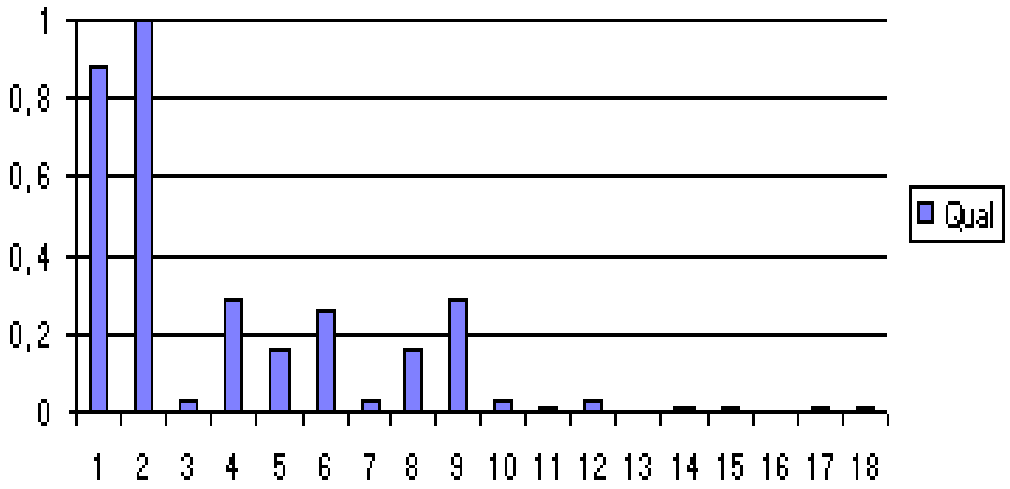

FIG. 6.9: *Participation à la qualité de prédiction pour chaque variable.*

A l'aide de ces résultats on détermine un seuil pour lequel on considère la variable comme "perturbatrice". Dans notre étude nous avons fixé le seuil à $\epsilon =$

0,01. On supprime donc les neurones de sortie i tel que :

$$I(V) \leq \epsilon$$

Appliqué à notre cas, les neurones "perturbateurs" sont : 11, 13, 14, 16 et 18. Dans le tableau 6.12, on peut remarquer que le fait d'avoir supprimé les neurones ci-dessus, a permis d'améliorer sensiblement les performances.

	Modélisation-18 87,93% [87,20%;88,63%]	
	BD	FD+ND
SN	51,53% 824	48,47% 775
SO	96,00% 1535	04,00% 64
SD	99,62% 1593	00,38% 06
SG	95,75% 1531	04,25% 68
SR	96,75% 1547	03,25% 52

	Modélisation-13 92,78% [92,19%;93,33%]	
	BD	FD+ND
SN	77,67% 1242	22,33% 357
SO	98,62% 1577	01,38% 22
SD	99,56% 1592	00,44% 07
SG	95,62% 1529	04,38% 70
SR	92,43% 1478	07,57% 121

TAB. 6.12: *Modélisation avec 18 variables vs Modélisation avec 13 variables*

En conclusion, la réduction de la dimension de l'espace de travail lors de la modélisation améliore les performances.

6.4 Conclusions

Dans ce chapitre, nous avons élaboré un nouveau système basé sur l'approximation de Bonferroni, appliqué aux modèles prédictifs connexionnistes.
La méthode d'approximation de Bonferroni et l'utilisation de la dernière couche cachée des réseaux comme modèle linéaire, nous a permis de calculer la région de confiance dans le cas vectoriel, où nous avons utilisé simultanément l'évolution de plusieurs indicateurs.
Nous avons de plus défini l'ébauche d'une nouvelle technique de sélection de variables, dans le cas de la combinaison de modèles prédictifs en vue d'identification. L'analyse de l'influence des variables sur la qualité de modélisation nous a permis de sélectionner un sous ensemble de variables maximisant le taux de bonnes détections et minimisant le taux de fausses détections. Cette approche permet en d'autres termes d'introduire une information "discriminante" dans la modélisation et par conséquent d'obtenir de bonnes performances de détection.

Chapitre 6. Bibliographie

[Bishop 1994] Bishop (C.). – Mixture Density Networks, Neural Computing Research Group Report. *NCRG/4288, Dept. of Comp. Sc., Aston University, Birmingham, UK*, 1994.

[Cibas et al. 1994] Cibas (T.), Fogelman Soulie (F.), Gallinari (P.) et Raudys (S.). – Variable Selection with Optimal Cell Damage. *ICANN'94*, 1994.

[Herrmann et al. 1989] Herrmann (F.), Stern (D.) et Chemouil (P.). – SUPERMAC: A Software Tool for the Performance Evaluation of Network Traffic Management. *ICCC, Symp. Beijing, China*, 1989.

[Jobson 1991] Jobson (J.D.). – Applied multivariate data analysis. *Regression and experimental design, Springer-Verlag*, vol. 1, 1991.

[Linde et al. 1980] Linde (Y.), Buzo (A.) et Gray (R.M.). – An Algorithm for the VQ Design. *IIIE, Trans. on Communication*, vol. 28, 1980, pp. 84–95.

[Nix et Weigend 1995] Nix (D.A.) et Weigend (A.S.). – Learning Local Error Bars for Nonlinear Regression. *Advances in Neural Information Processing Systems, NIPS7, MIT Press*, 1995, pp. 489–496.

[Stern et Chemouil 1992] Stern (D.) et Chemouil (P.). – A Diagnosis Expert System or Network Traffic Management. *Networks92, Kobe, Japan*, 1992.

[Stern 1991] Stern (D.). – A Statistical Study o Real-Time Telephone Traffic Variations or Network Management. *ITC Specialist Seminar, Krakw, Poland*, 1991.

[Thiria et al. 1992] Thiria (S.), Mejia (C.), Badran (F.) et Crepon (M.). – Multimodular Architecture for Remote Sensing Operations. *in lippmann R., Moody J.E., Touretzky (ed.) Neural Information Processing System*, vol. 4, 1992.

[Yacoub et Bennani 1997] Yacoub (M.) et Bennani (Y.). – HVS : A heuristic for variable selection in multilayer artificial neural network classifier. *Intelligent*

Engineering Systems Through Artificial Neural Networks, vol. 7, 1997, pp. 527–532.

Chapitre 7

Identification de Perturbations

Non naturel me contraint à chercher et aimer les choses bien ordonnées, fuyant la confusion qui m'est contraire et ennemie comme est la lumière des obscures ténèbres.

Nicolas POUSSIN (1594-1665).

L'identification est la deuxième tâche à accomplir dans un problème de diagnostic. Elle est essentielle pour renseigner sur le type d'incident survenu, et donne également des informations sur leur localisation. Dans ce chapitre, nous traitons ce problème, dans une première phase, à l'aide de combinaisons de modèles prédictifs. Après avoir présenté différentes techniques de combinaisons de modèles, nous avons étudié dans une deuxième phase, l'apport de la non-linéarité sur la linéarité en fusion de décision

7.1 Modélisation discriminante multivariée

7.1.1 Modèle prédictif pour l'identification

L'idée dans un modèle prédicitif est de minimiser l'erreur de prédiction associée à la série temporelle à traiter. Ce sont ces types de modèles que nous avons utilisés pour les études précédentes. Ce type de modèle n'intègre pas de discrimination car chaque modèle est indépendant de l'autre. L'information "interclasse" dans le cas où l'on associe une série à une classe n'est donc pas utilisée.

Chaque Réseau Connexionniste Prédictif (RCP) i modélise une fonction Ψ_i telle que :

$$X_t = \Psi(W_i, P_{t-1}^{t-d}) + e_{it} \quad \forall t = 1 \cdots T$$

avec

- T la taille d'une séquence
- $X_1^T = \{X_1, \cdots, X_T\}$ une sequence de vecteurs d'indicateurs,
- W_i représente l'ensemble des paramètres du modèle prédictif RCP_i,
- P_{t-1}^{t-d} le contexte de X_t (le passé),
- e_{it} un bruit blanc (erreur de prédiction).

Dans le cas d'une prédiction d'ordre d, $P_{t-1}^{t-d} = (X_{t-1}, \cdots, X_{t-d})$.

L'apprentissage d'un RCP se résume donc à déterminer W à partir d'un ensemble d'apprentissage S qui minimise, par exemple, la fonction de coût suivante :

$$J(W) = \sum_{k \in S} e_k^2$$

Le principe d'identification de l'anomalie pouvait donc se ramener à déterminer le modèle qui, pour un vecteur d'entrée donné à l'instant t, prédisait le mieux l'état du système à l'instant t+1.
Si on suppose que e_{it} suit une loi gaussienne de moyenne μ_i et de matrice de covariance Σ_i alors, (X_t / P_{t-1}^{t-d}) suit également une loi gaussienne $\aleph(\Psi_i(W_i, P_{t-1}^{t-d}) + \mu_i, \Sigma_i)$ et on a :

$$P_i(X_t / P_{t-1}^{t-d}) = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_i|}} e^{\left[-\frac{1}{2} D(x_t - \Psi(W_i, P_{t-1}^{t-d}))\right]}$$

avec m la dimension des vecteurs d'entrée (le nombre d'indicateurs) et

$$D(x_t - \Psi_i(W_i, P_{t-1}^{t-d})) = \left[x_t - \Psi_i(W_i, P_{t-1}^{t-d}) - \mu_i\right]^t \Sigma_i^{-1} \left[x_t - \Psi_i(W_i, P_{t-1}^{t-d}) - \mu_i\right]$$

La probabilité a posteriori de la situation S_i est donnée par :

$$P(S_i/X_t, P_{t-1}^{t-d}) = \frac{P_i(X_t/P_{t-1}^{t-d})}{\sum_{j=1}^{n} P_j(X_t/P_{t-1}^{t-d})}$$

où n est le nombre de situations (classes) étudiées.

Le fait de minimiser uniquement l'erreur de prédiction sur la série étudiée, ne nous garantit pas que l'erreur sera importante si l'on présente un vecteur d'une autre série (test).En fait si l'on apprend un RPC sur une situation S_i sûr et que l'on mesure l'erreur de prédiction E_i sur une sequence X_1^T (données observées sur cette situation S_i), on ne peut pas être sûr que l'erreur de prédiction E_j sur une autre sequence X'^T_1 (données observées sur une situation S_j) soit supérieure à E_i. Pour cela il faut, en plus de la minimisation de l'erreur de prédiction, maximiser cette erreur lorsqu'il s'agit d'un vecteur d'une autre série.

7.1.2 Modèlisation discriminante

Pour introduire la discrimination lors de la phase d'apprentissage des modèles RCP, nous avons procédé de la façon suivante :

Soient :

y_i^k : La sortie calculée de la $i^{ème}$ unité du RCP_k

d_i : La sortie désirée de la $i^{ème}$ unité, c'est-à-dire la valeur du $i^{ème}$ indicateur.

L'erreur de prédiction commise par le RCP_k est définie par :

$$E_k = \sum_i (y_i^k - d_i)^2$$

On considère

$$C_k = e^{-\sum_i (y_i^k - d_i)^2}$$

Notons p_k, l'erreur du prédicteur k normalisée sur l'ensemble des n RCP :

$$p_k = \frac{C_k}{\sum_{i=1}^{n} C_i}$$

Il en résulte que : $\sum_{k=1}^{m} p_k = 1$

Le principe de la méthode consiste donc à renforcer le modèle d'une perturbation sur les données de cette perturbation (le bon modèle : RCP_c) et éloigner des autres modèles. Ce type d'approche a donné de bons résultats en reconnaissance de la parole [Mellouk et Gallinari1993].
Soit p_c l'erreur normalisée de ce RCP_c, et p_k l'erreur normalisée des autres RCP_k avec $k \neq c$.

Le but du critère discriminant est de modifier les poids des modèles de façon à diminuer p_k et d'augmenter p_c.
L'algorithme d'apprentissage discriminant est le suivant :
Pour un vecteur d'indicateurs X_t donné d'une perturbation c et un RCP_c, la mise à jour des poids d'un RCP se fait par :

Pour tous les $RCP_z, z \in \{1, \cdots, n\}$ **faire**

  Pour tous les $j \in$ l'unité de sortie **faire**

    Si $(z = c)$ **alors**
      $\delta_j = f'(a_j) 2 p_c (y_j - d_i)(1 - p_z)$
    sinon
      $\delta_j = f'(a_j) 2 p_c (y_j - d_i)(-p_z)$
    Finsi

    $\Delta w_{ji} = \epsilon \delta_j y_j$

  Finpour

  Pour tous les $j \in$ l'unité cachée **faire**

    $\delta_j = f'(a_j) \sum_i \delta_i w_{ij}$
    $\Delta w_{ji} = \epsilon \delta_j y_j$

  Finpour

Finpour

Pour cette méthode, il faut noter que si $p_c = 1$, c'est-à-dire que le "bon" RCP a produit la meilleure prédiction (tous les autres p_k, $k \neq c$ sont nuls), aucune modification de poids n'est faite.
Par contre, si le "bon" RCP a produit une très mauvaise prédiction ($p_c = 0$) et qu'un autre RCP ait produit la meilleure ($p_k \approx 1$), le changement des poids se fera de façon à éloigner ce RCP_k et à rapprocher RCP_c.

7.1.3 Résultats et comparaisons

Il s'avère à travers les différents résultats obtenus que l'information inter-classe apporte un gain significatif au niveau des taux de bonnes détections. L'intégration de la discrimination que nous avons introduit dans notre système a été abordé dans l'apprentissage, comme une compétition entre les différents RCP.
Cependant, ce type d'approche n'égale pas le système basé sur une discrimination directe. On peut toutefois remarquer que notre système donne des résultats très proches de la discrimination directe (Cf. Tab. 7.1). En effet celle-ci obtient une performance moyenne de $81,25\%$ de bonnes classification, soit avec une probabilité de 95% l'intervalle de confiance $[80,41\%, 82,07\%]$. Ces résultats s'expliquent surtout du fait qu'il y ait une confusion entre deux états (Cf. Tab. 7.2). Cette confusion est du au fait que nos indicateurs reflettent essentiellement l'état d'un centre au niveau local et qu'en observant que celui-ci, il est trés difficile de différencier un surcharge régionale, d'une surcharge globale.

Modèles	Architectures	Intervalle de confiance à 95%
Prédictifs non discriminants	$< 18\|18\|18 >$	[76,42% , 80,24%]
Prédictifs discriminants	$< 18\|18\|18 >$	[77,92% , 81,65%]
Classifications directes	$< 18\|18\|5 >$	$[80,41\%, 82,07\%]$

TAB. 7.1: *Comparaison de l'approche discriminante, non- discriminante et classification directe.*

	SN	SO	SD	SG	SR
SN	**97.80%**	0.50%	0.90%	0.60%	0.60%
SO	2.50%	**96.50%**	0.00%	0.80%	0.20%
SD	2.80%	0.00%	**96.40%**	0.50%	0.30%
SG	9.50%	1.20%	0.00%	**67.70%**	48.70%
SR	18.00%	0.00%	0.30%	33.00%	**48.70%**

TAB. 7.2: *Matrice de confusion de la classification directe.*

7.2 Fusion de décisions & combinaison de modèles

Nous présenterons ici un ensemble de méthodes de combinaisons de multiples classifieurs, en l'occurence des perceptrons multi-couches afin de déterminer un

système de diagnostic performant et résistant au bruit.
L'idée de cette combinaison est que le pouvoir discriminatif de deux classifieurs non combinés est moindre qu'en fusionnant leurs décisions. Pour donner une notion intuitive de cette idée, prenons un exemple:
Si l'on prend deux experts, l'un spécialiste des problèmes occulaires que nous appelerons X , l'autre des problèmes auditifs appelé Y, si l'on demande un diagnostic de patients ayant pour 50% d'entre eux des problèmes occulaires et pour les 50% restant des problèmes auditifs.
On peut supposer que X fera 50% de diagnostic correct et qu'il en sera de même pour Y. Le résultat global sera assez médiocre, à savoir 50 % de bon diagnostic. Si à present, on demande aux deux experts un diagnostic pour chaque patient et que l'on prend comme réponse celui qui aura donné le bon diagnostic, on obtiendra de cette manière un résultat nettement supérieur, en tout état de cause 100%.
En résumé, l'idée sous-jacente de la combinaison est de créer des systèmes plus performants et plus résistants au bruit.

S'il est de pratique courante dans les modèles connexionnistes d'entraîner plusieurs réseaux différents et de sélectionner le meilleur d'entre eux en terme de performance, ce type de méthode n'est pas forcément très intéressant. En effet deux problèmes majeurs se posent pour cette méthodologie:

- L'évaluation du système connexionniste se faisant sur une base de validation, rien indique que le plus performant sur la base de validation se trouvera être le meilleur sur la base test, sauf s'il on dispose d'une base de validation "assez grande" taille très difficile à estimer.

- Comme nous l'avons vu précédemment, l'utilisation d'un seul système peut entraîner une perte importante des performances.

Une approche alternative à ce problème est la combinaison de multiples classifieurs. Il y a eu un grand intérêt à une telle approche et plusieurs études dans différents domaines de la reconnaissance des formes ont montré expérimentalement que la combinaison de multiples classifieurs permet de capturer des phénomènes complexes et d'améliorer les performances en classification [Bennani1995], [Ghosh et al. 1996] et [Guermeur et Gallinari1996].

7.2.1 Quelques techniques de combinaisons

Plusieurs méthodes de combinaison de classifieurs ont été suggérées :

7.2.1.1 Les méthode d'ensemble [Hansen et Salamon1990].

Cette technique repose sur l'hypothèse que chaque réseau de neurones entraîné pour la classification peut avoir convergé localement. Si bien qu'en utilisant plu-

sieurs réseaux, on peut espérer s'affranchir du problème de convergence. Ainsi, si les réseaux sont considérés comme indépendants, il est probable que ceux-ci ont convergé sur des minima locaux différents, et le fait de les combiner permettra d'obtenir une solution plus proche de la solution optimale. Pour cela, on peut utiliser un vote majoritaire afin d'obtenir une solution unique pour l'ensemble des architectures. Théoriquement, il a été prouvé qu'en procédant de cette manière, on augmente le pourvoir discriminant. Cependant, l'étude théorique se base sur le postulat d'indépendance des différentes architectures, ce qui est malheureusement rarement le cas.

7.2.1.2 Les méthodes de boosting [Drucker et al. 1993].

Ce système de combinaison repose sur trois réseaux (ou modèles) R_1, R_2, et R_3 dont les ensembles d'apprentissage seront construits de façon incrémentale à savoir : l'ensemble A_2 d'apprentissage dépendra des résultats de R_1 appris sur A_1, et enfin A_3 sera dépendant des réponses de R_1 et R_2 sur l'ensemble $A_1 \cup A_2$. Plus précisément, le protocole est le suivant :

- R_1 est entraîné sur l'ensemble A_1.
- Soit A'_1, le sous-ensemble de A_1 des x tel que ces x soient mal classés par R_1.
- On définit alors A'_2 tel que si $x \in A'_2$ alors $x \notin A_1$ et que $|A'_2| = |A'_1|$ avec $|.|$ correspondant à la cardinalitè de l'ensemble, c'est-à-dire que l'ensemble A'_2 a comme nombre d'élément, le même nombre d'éléments mal classé par R_1. On obtient ainsi $A_2 = A'_1 \cup A'_2$, ensemble d'apprentissage de R_2.
- On construit enfin A_3 tel que si $x \in A_3$, alors R_1 et R_2 ne donnent pas la même réponse pour x.

L'utilisation des trois réseaux est alors la suivante: pour un exemple x on le présente à R_1 et à R_2. Si ceux-ci donnent la même réponse, on classe x suivant leurs réponses. Sinon on présente x à R_3 et on choisit la réponse de R_3. Cette technique a été validée sur des problèmes de la reconnaissance de caractères, et semble donner de très bons résultats.

7.2.1.3 les techniques d'empilement « stacking » [Wolpert1992].

Cette technique peut être vue comme une généralisation du système de validation croisé. Elle se décompose comme suit :

- On dispose d'un ensemble E (apprentissage, validation) et de k réseaux (ou modèle) R_k.

- On définit z ensembles d'apprentissages A_i et de validations V_i tels que $A_i \cup V_i = E, \forall i \in \{1, \cdots, z\}$.

- On entraîne les k réseaux sur les z ensembles d'apprentissage. Cela nous donne kz réseaux, où R_k^z correspond au réseau R_k ayant appris sur l'ensemble d'apprentissage A_z .

- Pour tous les ensembles de validations $V_i, i \in \{1, \cdots, z\}$, faire

  - Pour tous les exemples $x \in V_i$

    Pour tous les réseaux $R_j^i, j \in \{1, \cdots, k\}$, déterminer la sortie $y_j^i(x)$.

    Ceci nous donne pour l'exemple x, un vecteur de k composants: $Y^i(x) = [y_1^i(x), \cdots, y_k^i(x)]$.

    on ajoute le couple $(Y^i(x), y)$ à un ensemble E' (y correspond à la sortie désirée pour l'exemple x).

    fin pour

  - fin pour

- On apprend à un nouveau réseau R_1, l'ensemble E'. On considère ce réseau de niveau 1.

En effet, celui-ci n'apprend plus sur les exemples, mais sur les sorties des réseaux afin de permettre de supprimer le biais. Il faut noter qu'il est tout à fait possible de recommencer les opérations ci-dessus sur l'ensemble E' pour obtenir un réseau de niveau 2. Les résultats obtenus avec ce type de méthode sont très bons sur la reconnaissance de caractères.

7.2.1.4 Les architectures multi-modulaires [Hampshire et Waibel1992], [Bennani1992] et [Lamy et Fogelman1995].

Dans cette approche, on spécialise chaque réseau (ou modèle) dans une tâche bien précise. On peut prendre comme exemple l'identification du locuteur où chaque réseau se spécialise sur un locuteur. L'idée est donc de créer un méta-module qui apprendra à minimiser l'erreur de la combinaison non-linéaire des différents réseaux. Ce méta-module a comme entrée le même signal que les autres réseaux mais, a comme fonction de coût :

- y_n : sortie globale du système pour l'exemple n.

- w_i : poids affecté au réseau i du système.

- s_n^i : sortie du réseau i pour l'exemple n.

$$J = ||y_n - \sum_i w_i s_n^i||^2$$

Cette approche permet de prendre en compte les réponses des différents réseaux, ces informations aui ne seraient pas prises en compte par un module superviseur qui n'apprendrait à faire confiance à un seul réseau.

7.2.1.5 Les systèmes multi-expert [Bennani1993], [Jacobs et al. 1991].

Ces techniques sont très proches de celles présentées ci-dessus. Cependant, ces dernières possèdent un expert pour chaque classe, les réseaux se spécilisant automatiquement. Ces techniques permet d'obtenir la probabilité de chaque expert sur une forme donnée, ce qui permet d'accentuer la localité de ceux-ci. De plus, un superviseur permet de prendre la décision finale.

Les techniques présentées ci-dessus sont plus finement décrites dans [Lamy1995].

Dans [Xu et al. 1992] on peut trouver une très riche synthèse de ces différentes méthodes de combinaison. Un cadre analytique de combinaison de classifieurs est fourni dans l'article de Ghosh [Tumer et Ghosh1996].

7.2.2 Fusion des décisions de multiples classifieurs

Dans cette étude, nous proposons d'employer une combinaison de classifieurs pour la détection et l'identification d'anomalies.
La combinaison de classifieurs peut être vue comme un ensemble d'experts regardant le même problème de leurs points de vue individuels et énonçant leurs opinions pour l'état actuel du réseau.

Dans le cas où des classifieurs sont entraînés pour une même tâche, on pourrait imaginer que la fusion de décision n'est pas utile.
En fait il n'en n'est rien. Chaque réseau étant entraîné indépendamment de l'autre, il y a de grandes chances que la généralisation des différents modèles connexionnistes soient complémentaires, d'où l'utilité de la combinaison.
Chaque classifieur individuel $R_k\,(k = 1, \cdots, K)$ produit une décision (sortie) $O_k(x \in C_i|x)$ pour chaque classe $C_i\,(i = 1, \cdots, M)$ quand une forme x est présentée.
La tâche centrale est la combinaison des différents votes O_k en un vote consolidé O_{comb}. Il existe plusieurs manières pour combiner le pouvoir discriminant des différents classifieurs. Dans cette étude nous considérons deux types de combinaison :

- linéaire,
- non-linéaire

7.2.2.1 La combinaison linéaire

La méthode la plus répandue consiste à combiner les sorties de k classifieurs sous forme d'une moyenne pondérée. Cette approche simple emploie la valeur suivante comme une nouvelle estimation de la sortie du système :

$$O_{comb}(x \in C_i|x) = \sum_{k=1}^{K} \omega_k O_k(x \in C_i|x)$$

où les valeurs optimales pour les ω_k peuvent être déterminées par minimisation de l'erreur quadratique moyenne sous la contrainte :

$$\sum_{k=1}^{K} \omega_k = 1$$

La solution peut être obtenue en utilisant les multiplicateurs de Lagrange :

$$\omega_k = \frac{\sum_{j=1}^{K} C_{kj}^{-1}}{\sum_{i=1}^{K}\sum_{j=1}^{K} C_{ij}^{-1}}$$

où C_{ij} de dimension (k,k) est la matrice de corrélation estimée en utilisant les données d'apprentissage et définie par :

$$C_{ij} = \frac{1}{N}\sum_{n=1}^{N}(O_i(x^n) - d^n)(O_j(X^n) - d^n)$$

avec

- $O_i(x^n)$ la sortie calculée par le classifieur $M_i\,(i = 1, \cdots, K)$ quand une forme x^n est présentée,
- d^n la sortie désirée correspondant à l'entrée x^n.

7.2.2.2 La combinaison non-linéaire

Cette approche peut être vue comme une méta-classification. La combinaison est faite d'abord par la concaténation des vecteurs de sorties (de dimension T=KxM, ou K est le nombre de réseaux et M le nombre de classes) des différents classifieurs.
Cette concaténation produit une nouvelle base de formes qui sera utilisée par la suite pour une tâche de classification.

Dans cette étude nous avons utilisé un réseau multi-couches pour effectuer cette méta-classification.
Cette méta-classification produit une combinaison non-linéaire des différentes sorties des classifieurs et peut être obtenue comme suit (Dans le cas d'un réseau à une couche cachée):

$$O_{comb}(x \in C_i|x) = f\left(\omega_{io} + \sum_j \omega_{ij} f\left(\omega_{jo} + \sum_{t=1}^{T} \omega_{jt} O_t(x)\right)\right)$$

où f est une fonction non-linéaire (sigmoïd), les valeurs optimales pour les w_{ij} peuvent être déterminées par apprentissage (minimisation de l'erreur quadratique moyenne par une procédure de gradient).
Dans les deux méthodes précédentes de combinaison, la décision finale est donnée par :

$$C_i^* = \underset{1 \leq i \leq M}{argmax}(O_{comb}(x \in C_i|x))$$

7.2.3 Application à la tâche de diagnostic

Soit $X_1^T = \{X_1, \cdots, X_T\}$ une séquence de taille T de vecteurs d'indicateurs et $P_{t-1}^{t-d} = (X_{t-1}, \cdots, X_{t-d})$ un contexte d'ordre d.

Dans cette étude, nous avons employé plusieurs modèles travaillant en parallèle et recevant la contribution de différents contextes.
Le modèle R(P_{t-1}^{t-d}) est un modèle de type PMC qui reçoit comme entrée la séquence de vecteurs d'indicateurs : $(X_{t-1}, \cdots, X_{t-d})$.

7.2.3.1 La détection de perturbations

Pour cette première tâche, nous avons généré une base de données composée de deux classes : la situation nominale et la situation perturbée, les vecteurs d'indicateurs étant de dimension 18. Nous avons choisit les contextes P_{t-2}^{t-1} et P_{t-2}^{t} afin d'obtenir une modélisation différente pour les deux réseaux afin qu'ils ne fassent pas le même type d'erreurs. La table 7.3 montre les résultats de détection pour chaque classifieur dépendant du contexte, .

Modèle	Architecture	Intervalle de Confiance à 95 %
R(P_{t-2}^{t-1})	$< 36\|20\|2 >$	90.10% [88.71 , 91.33]
R(P_{t-2}^{t})	$< 54\|30\|2 >$	91.67% [90.18 , 92.94]

TAB. 7.3: *Les performances de modules individuels pour la tâche de détection.*

Tous ces résultats sont donnés sous forme de performance moyenne et un intervalle de confiance à 95 %.

Méthode de Combinaison	Intervalle de Confiance à 95 %
Linéaire	91.80% [90.32 , 93.06]
Non-linéaire	91.80% [90.32 , 93.06]

TAB. 7.4: *La comparaison de méthodes de combinaison pour la tâche de détection.*

Des tables 7.3 et 7.4, on peut remarquer que la combinaison dans tous les cas donne les meilleurs résultats. Cette réduction d'erreur est due à la réduction de la variance par combinaison de votes.
En outre, nous pouvons noter que pour ce problème à deux classes, les deux types de combinaison donnent les mêmes performances.

7.2.3.2 L'identification de perturbations

Tous les classifieurs sont entrainés à discriminer entre les 5 types de situations étudiées.

Modèle	Architecture	Intervalle de Confiance à 95 %
$R(P_{t-2}^{t-1})$	$< 36\|20\|5 >$	85.00% [83.37 , 86.50]
$R(P_{t-2}^{t})$	$< 54\|30\|5 >$	86.37% [84.57 , 87.99]

TAB. 7.5: *Les performances de modules individuels pour la tâche d'identification.*

La table 7.5 montre que les performances des différents classifieurs dépendent du contexte utilisé. Nous avons alors essayé de fusionner les décisions de ces deux modèles : $R(P_{t-2}^{t-1})$ et $R(P_{t-2}^{t})$.

Méthode de Combinaison	Intervalle de Confiance à 95 %
Linéaire	89.41% [87.77 , 90.84]
Non-linéaire	89.53% [87.91 , 90.96]

TAB. 7.6: *La comparaison de méthodes différentes de combinaison pour la tâche d'identification*

Des deux tables 7.5 et 7.6, nous pouvons observer, en comparaison de performances avec les modules individuels, que l'approche par combinaison améliore les performances. On peut aussi remarquer que la règle de combinaison non-linéaire est meilleure que la méthode linéaire dans ce cas difficile de classification.

7.3 Conclusion

Nous avons proposé des systèmes basés sur la combinaison de différents modèles connexionnistes, permettant d'obtenir de bonnes performances de détection et d'identification.
Ces systèmes sont modulaires et permettent l'addition d'autres modèles. Les motivations de la combinaison de classifieurs dans cette étude sont :

- l'amélioration de performances,
- la parallélisation,
- la résistance aux pannes (la destruction d'un module de classifieur n'implique pas la modification du système entier).

Il faut noter que cette coopération de modèles provoque un accroissement de la complexité de calcul :

- la combinaison multiplie la complexité (en coût de calcul).
  Le choix d'un système combinant plusieurs modèles devra être fait par un compromis entre:
  - l'efficacité,
  - la complexité.
- la diminution de l'erreur par la combinaison dépend de l'indépendance des modèles employés dans la combinaison.

Chapitre 7. Bibliographie

[Bennani 1992] BENNANI (Y.). – Approche connexionnistes pour la Reconnaissance Automatique du Locuteur : Modèlisation et Identification. *Thèse à l'université de Paris-Sud centre d'Orsay*, 1992.

[Bennani 1993] BENNANI (Y.). – Probabilistic Cooperation Of Connectionist Expert Modules: Validation On A Speaker Identification Task. *IEEE , ICASSP'93, Minneapolis, Minnesota, U.S.A.*, 1993.

[Bennani 1995] BENNANI (Y.). – Modular and Hybrid Connectionist System for Automatic Speaker Identification. *Neural Computation, MIT Press*, vol. 7(4), 1995, pp. 791–797.

[Drucker et al. 1993] DRUCKER (H.), SCHAPIRE (R.) et SIMARD (P.). – Boosting Performance in Neural Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7(4), 1993, pp. 705–719.

[Duda et Hart 1973] DUDA (R.O.) et HART (P.E.). – Pattern classification and scene analysis. *Wiley, New York*, 1973.

[Ghosh et al. 1996] GHOSH (J.), BECK (S.) et DEUSER (L.). – Integration of Neural Classifiers for Passive Sonar Signals. *C.T. Leondes, editor, DSP Theory and Applications, Academic Press.*, 1996.

[Guermeur et Gallinari 1996] GUERMEUR (Y.) et GALLINARI (P.). – Combining Statistical Models for Protein Secondary Structure Prediction. *ICANN'96*, 1996.

[Hampshire et Waibel 1992] HAMPSHIRE (J.B.) et WAIBEL (A.H.). – The Meta-Pi Network: Building Distributed Representations for Robust Multisource Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(7), 1992, pp. 751–769.

[Hansen et Salamon 1990] HANSEN (L.K.) et SALAMON (P.). – Neural Network Ensembles. *Transactions on Pattern Analysis and Machine Intelligence*, 1990, pp. 993–1001.

[Jacobs et al. 1991] JACOBS (R.A.), JORDAN (M.I.), NOWLAN (S.J.) et HINTON (G.E.). – Adaptive Mixtures of Local Experts. *Neural Computation*, vol. 3, 1991, pp. 79–87.

[Lamy et Fogelman 1995] LAMY (B.) et FOGELMAN (F.). – Can Multiple Initializations Improve Generalization? *ICANN'95*, vol. 1, 1995, pp. 39–44.

[Lamy 1995] LAMY (B.). – Reconnaissance de caractères manuscrits par combinaison de modèles connexionnistes. *Thèse de doctorat l'université de Paris-6*, 1995.

[Mellouk et Gallinari 1993] MELLOUK (A.) et GALLINARI (P.). – A discriminative neural prediction system for speech recognition. *ICASSP'93, Minneapolis, USA.*, 1993.

[Tumer et Ghosh 1996] TUMER (K.) et GHOSH (J.). – Theoretical Foundations of Linear and Order Statistics Combiners for Neural Pattern Classifiers. *TR-95-02-98, CVRC, University of Texas, Austin*, 1996.

[Wolpert 1992] WOLPERT (D.H.). – Stacked Generalization. *Neural Computation, MIT Press*, vol. 5(2), 1992, pp. 241–259.

[Xu et al. 1992] XU (L.), KRZYZAK (A.) et SUEN (C.Y.). – Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *Transactions on Systems, Man and Cybernetics*, vol. 22(3), 1992, pp. 418–435.

Chapitre 8

Conclusion et Perspectives

Se sens que je progresse
à ceci que je recommence
à ne rien comprendre à rien.

Charles Ferdinand Ramuz (1878-1947).

8.1 Conclusion et Perspectives

Dans cette thèse, nous nous sommes intéressés à l'utilisation et l'adaptation des techniques connexionnistes dans la réalisation des systèmes de diagnostic de systèmes complexes. Nous avons d'abord abordé le problème difficile de la sélection de variables. Bien que ce soit un vieux problème de statistique, il est toujours ouvert. Nous avons proposé deux nouvelles mesures de pertinence permettant de quantifier l'importance de chaque variable dans le système d'apprentissage. Ces techniques d'élagage permettent d'une part, d'ajuster la complexité du modèle à la difficulté du problème et d'autre part, de sélectionner un sous-ensemble de caractéristiques pertinentes. Les deux méthodes que nous avons proposées sont comparées avec d'autres approches de sélection de variables connexionnistes et conventionnelles parmi les plus connues. L'efficacité de nos méthodes a été testée sur plusieurs problèmes. Nous avons observé dans nos expériences que nos méthodes permettent une meilleure détection des dépendances non-linéaires entre les variables que les techniques conventionnelles. Nous avons ensuite abordé le problème de diagnostic du réseau téléphonique en utilisant les techniques de sélection de variables comme pré-traitement des données. Plus précisément, nous avons utilisé les méthodes de sélection de variables pour choisir les sous-ensembles de caractéristiques (indicateurs) pertinentes pour la génération d'alarmes et l'identification de perturbations. Nous avons proposé plusieurs systèmes connexionnistes multi-modulaires pour réaliser d'une part, des tâches de diagnostic sur le réseau téléphonique visant à assurer la détection d'incidents et d'autre part, l'identification de perturbations. Nous avons abordé ces deux problèmes du diagnostic (détection et identification) avec deux approches différentes :

- l'approche par modélisation connexionniste:
  dans ce cas, nous avons proposé des modèles du système dans ces différents modes. Ces modèles calculent dans le cas univarié un intervalle de prévision qui est ensuite utilisé pour la détection et dans le cas multivarié, ces modèles calculent des régions de confiance utilisées pour la détection. Nous avons ensuite proposé une heuristique permettant de sélectionner les meilleures régions de confiance afin d'améliorer les résultats de détection.

- l'approche par combinaison de modèles:
  dans ce cas, nous avons étudié deux possibilités de fusion de décision. La première est linéaire et consiste à pondérer les réponses de chaque modèle. La deuxième est une technique de combinaison non-linéaire qui consiste à apprendre par un réseau connexionniste les coefficients de pondération des décisions de chaque modèle.

Dans cette étude, nous avons travaillé sur les diagnostics locaux en considérant les variables d'états du trafic téléphonique correspondant au CTS (ou au CTP) afin de « diagnostiquer » le type de situation dans laquelle est ce centre. Une

perspective consiste à fusionner les informations provenant des différents centres locaux pour faire un diagnostic global du réseau. Une dernière perspective consiste à faire coopérer les techniques de sélection de variables et les approches de fusion de décisions dans un cadre global.

Bibliographie personnelle

[Bennani et al. 1996] BENNANI (Y.), BOSSAERT (F.), LERAY (P.) et GALLINARI (P.). – Architectures Neuronales pour la Détection et la Classification de Perturbations dans un Réseau Téléphonique. *Actes des Séminaires Action Scientifique-CTI: Diagnostic des Systèmes Complexes, Issy les Moulineaux*, 1996.

[Bennani et al. 1997] BENNANI (Y.), BOSSAERT (F.) et DIDELET (E.). – Linear and Nonlinear Combinations of Connectionist Models for Local Diagnosis. *International Conference on Artificial Neural Networks (ICANN'97), Lausanne Suisse*, 1997.

[Bennani et Bossaert 1995] BENNANI (Y.) et BOSSAERT (F.). – A Neural Network Based Variable Selector. *International Conference on Artificial Neural Networks and Intelligent Engineering, (ANNIE'95), S^t Louis, Missouris USA*, 1995.

[Bennani et Bossaert 1996] BENNANI (Y.) et BOSSAERT (F.). – Predictive Neural Networks for traffic Disturbance Detection in the Telephone Network. *Computational Engineering in Systems Application (CESA'96), IEEE-SMC, Lille*, 1996.

[Bennani et Bossaert 1998] BENNANI (Y.) et BOSSAERT (F.). – Alarm Generation in Telephone Network Using Multivariate Neural Networks Modelling Join Confidence Interval. *International Workshop on Principles of Diagnosis (DX'98), Cape Cod, Massachusetts USA*, 1998.

[Bossaert et Benjamin 1999] BOSSAERT (F.) et BENJAMIN (D.). – AINS: Architecture Independent Neural Selection. *International Joint Conference on Neural Networks (IJCNN'99), Washington USA*, 1999.

[Bossaert et Bennani 1996] BOSSAERT (F.) et BENNANI (Y.). – From Implicit to Explicit Information in Multilayer Neural Networks. *International ICSC on Intelligent Industrial Automation and Soft Computing, (ICSC'96), Reading UK*, 1996.

Chapitre 9

Annexes: Analyse descriptive des données téléphoniques

9.1 ACP

L'analyse en composantes principales permet de réaliser des combinaisons linéaires des variables d'entrée et de déterminer ainsi une nouvelle base pour représenter les données. Les vecteurs de cette base (appelés axes principaux) sont choisis séquentiellement de façon à réaliser une base orthogonale et à maximiser la dispersion des données lors de la projection sur chacun de ces axes. Cette technique d'analyse des données est bien décrite dans [Saporta1978].
En analyse exploratoire des données, on utilise des projections permettant une visualisation, i.e. on se restreint à deux ou trois axes. Pour les données étudiées, les deux premiers axes permettent une bonne visualisation.
Nous commencerons par faire une ACP générale sur l'ensemble des données en regardant aussi si les bases d'apprentissage et de test sont bien représentatives. Ensuite nous ferons une ACP pour certaines surcharges afin de regarder la répartition des données à l'intérieur d'une classe.

9.1.1 ACP générale

Pour les surcharges faibles (proches de 50 %), on distingue un regroupement des différentes classes autour de l'état nominal. Plus les surcharges augmentent plus les classes s'écartent (au sens distance euclidienne du terme). Notons aussi que pour chaque classe on observe des taches (ou groupements) distinctes qui correspondent aux différents taux de surcharge.
Les différentes situations sont bien séparées et quelques axes de l'ACP permettent d'expliquer une partie importante des données (cf. tables 9.1 et 9.2). Les valeurs observées pour les valeurs propres et l'inertie montrent que la dimension "linéaire" du problème est de 4 (l'inertie cumulée de la 4^ieme valeur propre est très proche

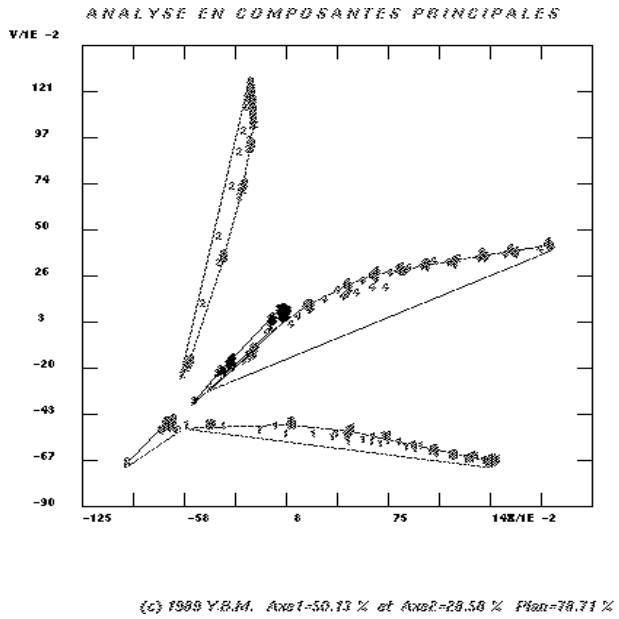

FIG. 9.1: *ACP sur la base d'apprentissage.*

de 100%). Bien que l'on n'ait pas fait de simulations pour de telles valeurs, on peut penser que les classes sont mélangées pour des taux de perturbation inférieurs à 50 %. Nous allons voir qu'il est simple de classer de telles données en différents types de perturbation. Les ensembles test et apprentissage des différentes perturbations apparaissent très similaires pour les scénarios que nous avons testés.

9.1.2 ACP par classe

Nous avons réalisé une série d'ACP pour différentes classes de perturbations. Nous montrons ci dessous les projections obtenues pour les surcharges origine

Valeur propre	Inertie	Cumul
9.023115	50.13	50.13
5.145047	28.58	78.71
2.339184	13.00	91.71
1.300397	7.22	98.93
0.090057	0.50	99.43
0.076829	0.43	99.86
0.024456	0.14	99.99
0.000862	0.00	100.00

TAB. 9.1: *Valeurs propres et part d'inertie des axes correspondants pour l'ensemble d'apprentissage .*

Valeur propre	Inertie	Cumul
9.124897	50.69	50.69
5.038583	27.99	78.69
2.434279	13.52	92.21
1.305884	7.25	99.46
0.086343	0.48	99.94
0.008937	0.05	99.99
0.000732	0.00	100.00

TAB. 9.2: *Valeurs propres et part d'inertie des axes correspondants pour l'ensemble de test .*

(fig. 9.2) et destination (fig. 9.3). Elles montrent le même comportement. On peut observer les paquets de données correspondant aux différents taux de surcharge simulés. La séparation n'est toutefois pas si nette car des données correspondant à différents taux de surcharge se trouvent mélangées. Cela peut être dû aux montées en charge des perturbations qui sont proches quelque soit le taux de surcharge simulé.
Une conséquence est que l'étiquetage des données en différents taux de surcharge est non trivial. Il faudra en tenir compte si l'on veut s'intéresser à la prédiction du taux de surcharge. Le tableau 9.3 donne les parts d'inertie de chaque axe et montre la faible dimensionalité intrinsèque de cet ensemble de données.

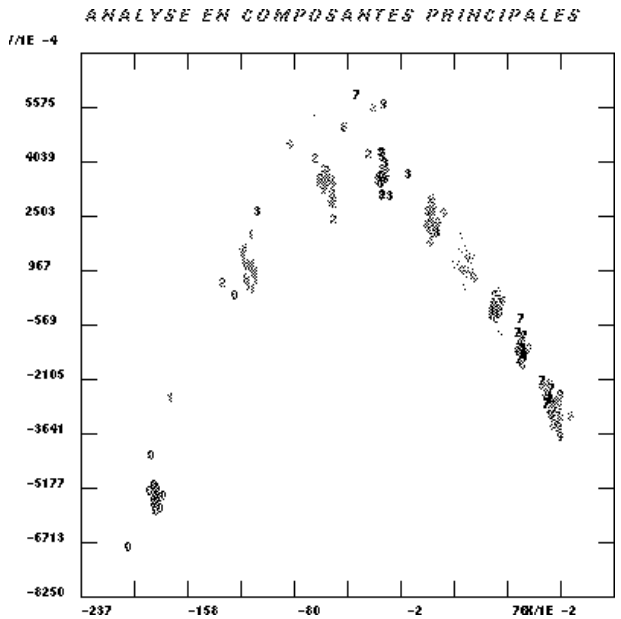

FIG. 9.2: *Surcharge origine.*

1 Saporta G. (1978), Théories et Méthodes de la Statistique, Éditions Technip.

Valeur propre	Inertie	Cumul
88.77	88.77	15.977948
8.79	97.56	1.582197
1.78	99.34	0.320305
0.40	99.73	0.071620
0.17	99.91	0.031345
0.07	99.98	0.012744
0.02	99.99	0.002795
0.01	100.00	0.001025

TAB. 9.3: *Valeurs propres et inertie des axes correspondants pour la surcharge origine.*

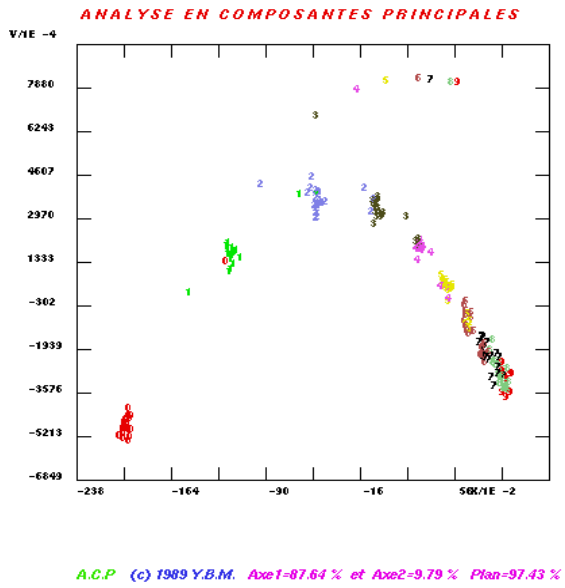

FIG. 9.3: *Surcharge destination.*

9.2 Étude de la montée en charge

Pour étudier le comportement de la montée en charge du réseau en vue de la prédiction du taux de surcharge d'une perturbation, nous avons dégagé deux points :

- A-t-on la possibilité de distinguer les différents taux d'une même surcharge ?
- Peut-on prédire le taux de surcharge lors de la montée en charge ?

Valeur propre	Inertie	Cumul
87.64	87.64	15.775094
9.79	97.43	1.761814
2.31	99.73	0.415172
0.14	99.87	0.024993
0.09	99.97	0.016886
0.02	99.98	0.003140
0.01	99.99	0.001946
0.01	100.00	0.000940

TAB. 9.4: *Valeurs propres et inertie des axes correspondants pour la surcharge destination.*

9.2.1 Distinction de différents taux d'une même surcharge

Les figures 9.4 , 9.5 et 9.6 représentent des ACP sur les données correspondant à un type de surcharge (origine, destination, globale), chaque classe représentant un taux Ti de surcharge (50 = Ti = 900 pour origine et destination et 25 = Ti = 100 pour globale). L'analyse de ces trois projections montre des regroupements intra-classes assez nets. La part d'inertie du plan de projection est d'environ 99 %.

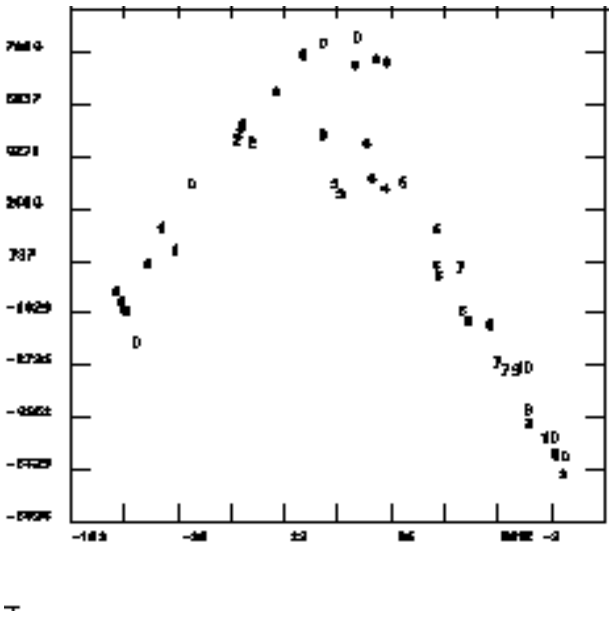

FIG. 9.4: *ACP surcharge origine .*

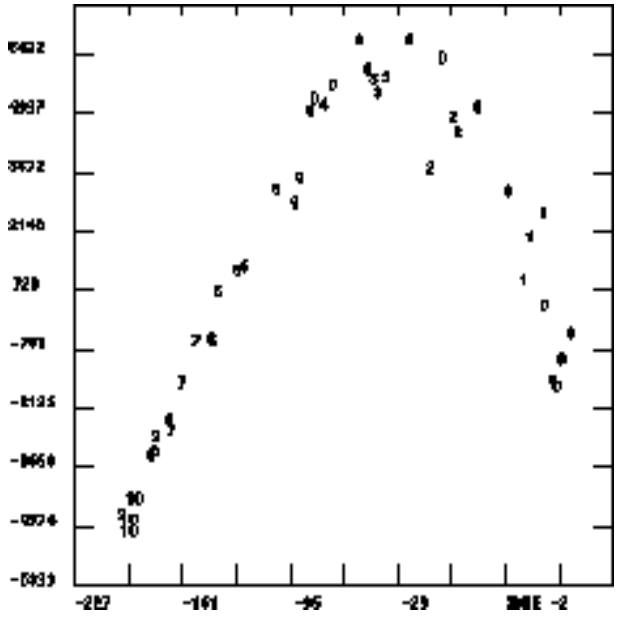

FIG. 9.5: *ACP surcharge destination.*

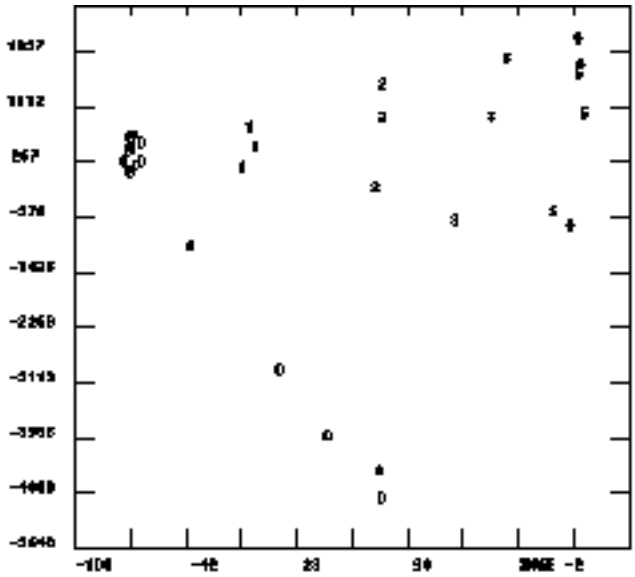

FIG. 9.6: *ACP surcharge globale.*

9.2.2 Prédiction du taux de surcharge lors de la montée en charge.

Pour ce second point, nous avons étudié l'évolution au cours du temps de différentes variables pour chaque surcharge, l'incident étant déclenché à la 45ème

minute (c'est à dire entre les mesures 11 et 12). Nous présentons ici plusieurs cas représentatifs, l'étude des autres variables se trouve en annexe.

9.2.2.1 Variable Appels efficaces origine pour la surcharge

origine, Variable Appels efficaces pour la surcharge destination.

Les figures 9.7 et 9.8 représentent respectivement l'évolution temporelle des variables, Appels efficaces origine et Appels efficaces . Sur chaque figure, une courbe représente un taux de surcharge Ti (50 % =Ti = 900 %, par pas de 50 %).

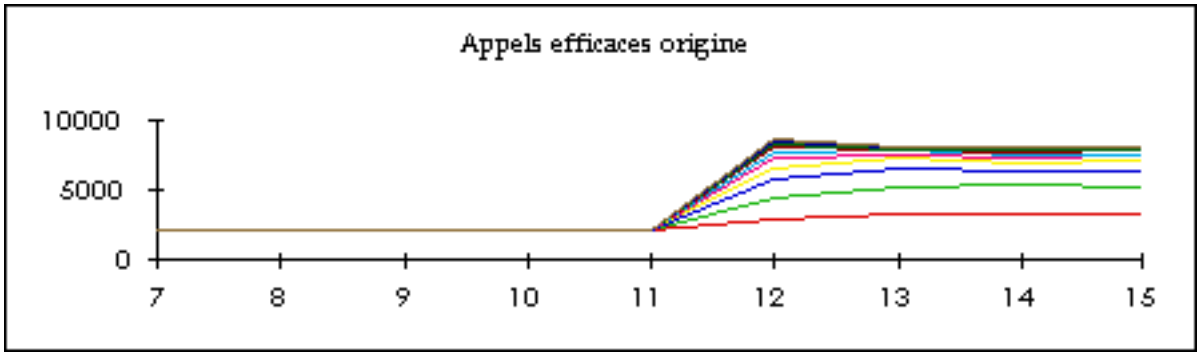

FIG. 9.7: *Évolution temporelle de la variable Appels efficaces origine.*

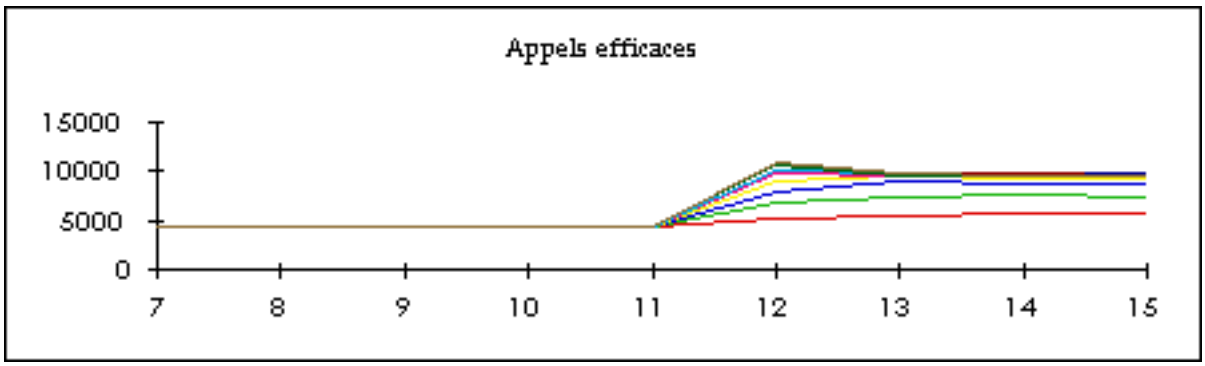

FIG. 9.8: *Évolution temporelle de la variable Appels efficaces.*

On peut remarquer sur les deux figures que, quelque soit le taux Ti, il y a une brusque transition entre les instants 11,12 et 13. Cela signifie que l'on passe généralement d'un état nominal à un état de surcharge au bout d'une mesure. Ce phénomène rend impossible la prédiction du taux de surcharge pendant la montée en charge.

1	tentatives
2	prises
3	prises inefficaces réseaux
4	prises efficaces
5	appels efficaces
6	trafic écoulé
7	trafic efficace
8	occupation totale

TAB. 9.5: *Liste des variables faisceaux utilisées.*

9.2.2.2 Variable Prises efficaces origine pour la surcharge globale

Chaque courbe de la figure 9.9 représente un taux de surcharge Ti (25% =Ti = 100%, par pas de 25%).

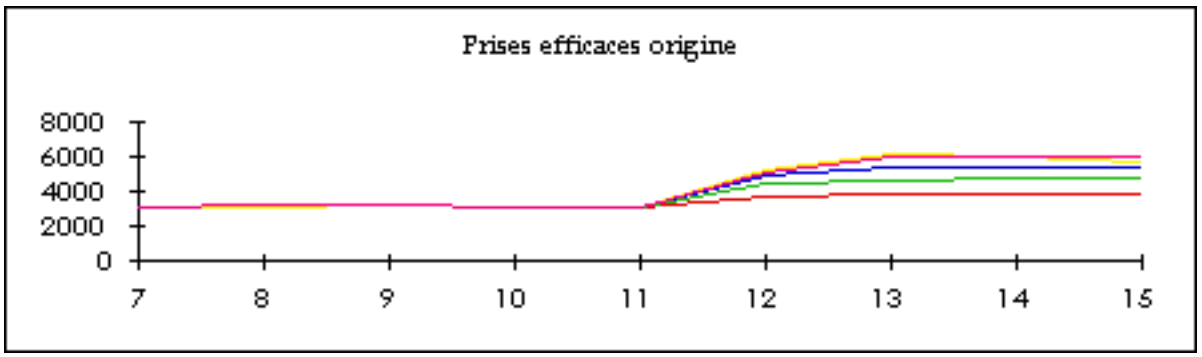

FIG. 9.9: *Évolution temporelle de la Variable Prises efficaces origine pour la surcharge globale.*

On peut remarquer sur cette figure une transition plus lente de l'état nominal à un état de surcharge. Ainsi la prédiction du taux de surcharge pourrait avoir lieu lors de la 12ème mesure.

9.2.3 Étude des indicateurs faisceaux au niveau d'un centre

Les variables que nous avons utilisées pour les faisceaux sont: Pour chaque mesure nous avons donc 41 variables : 25 pour le centre, 8 pour les faisceaux entrants et 8 pour les faisceaux sortants.

Les scénarios que nous avons utilisés sont :

- État nominal

  10 scénarios avec des variables aléatoires différentes d'où 190 mesures.

- Surcharge Origine

10 scénarios avec une surcharge Ti (50 % =Ti = 900 %, par pas de 50 %).

- Surcharge Destination

  10 scénarios avec une surcharge Ti (50 % =Ti = 900 %, par pas de 50 %).

- Surcharge Globale

  10 scénarios avec des taux de surcharges :

  25 %,50 %,100 %,150 %,200 % et pour chaque taux de surcharge un deuxième scénario avec une variable aléatoire différente.

- Surcharge Régionale

  10 scénarios avec un taux de surcharge Ti (25 % =Ti = 125 %, par pas de 25 %) et pour chaque taux de surcharge un deuxième scénario avec une variable aléatoire différente.

- Jeux téléphoniques

  10 scénarios avec des variables aléatoires différentes.

L'ensemble de ces scénarios nous fournit une base de 990 exemples.

Nous avons ensuite supprimé les variables toujours nulles, il reste les 18 variables 'centre' habituelles et 15 des 16 variables 'faisceaux' (prises inefficaces sortant est nulle)

Pour déterminer l'utilité des variables 'faisceaux', nous avons procédé à deux ACP, la première sur les variables 'centre' seulement, la seconde sur l'ensemble des variables 'centre' et 'faisceaux'. Les figures 9.10 et 9.11 sont très ressemblantes, ce qui nous permet seulement de dire que les variables 'faisceaux' ne sont pas d'une importance fondamentale pour notre étude. Cependant, nous en conserverons certaines pour nos prochaines études pour plusieurs raisons :

- Certaines variables 'faisceaux' peuvent être plus parlantes pour les experts.
- Les mesures réelles de certaines de ces variables sont disponibles (données VIOLETTE).

9.2.4 Conclusion sur l'analyse des données

Pour une période d'échantillonnage de 4 minutes, cette étude nous montre que la montée en charge, du réseau est représentée par une mesure voire aucune. Le calcul du taux de surcharge est possible dès la première mesure car la transition état nominal - surcharge est très peu visible. Pour parler de prédiction du taux de surcharge pendant la montée en charge il faudrait prendre une période d'échantillonnage plus petite, les indicateurs représentatifs d'un centre arrivant toutes les 15 secondes.

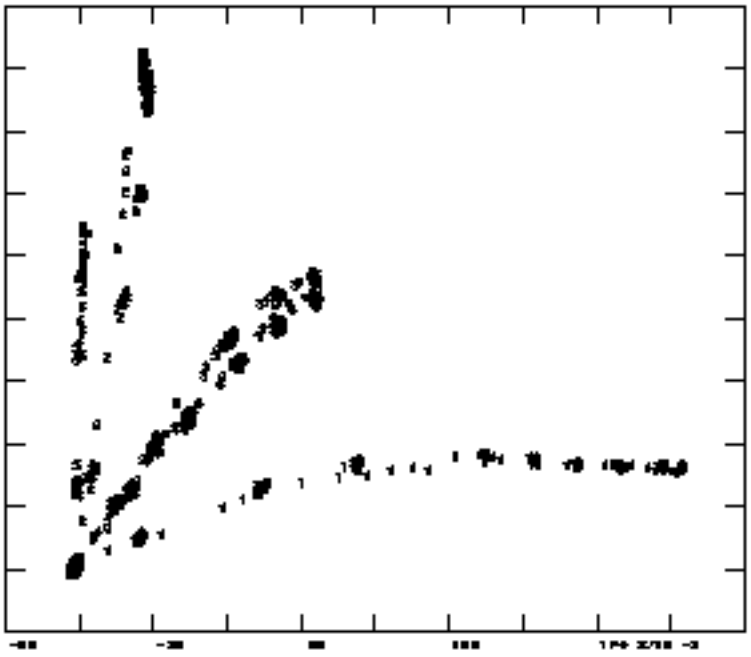

FIG. 9.10: *ACP des données relatives aux variables du centre.*

9.2.5 Bases de données

Nous avons utilisé le simulateur SuperMac, pour générer la base de données. L'étude des précédents problèmes nous a permis de déterminer deux indicateurs majeurs à savoir :

- IS (Incoming Seazures)
- OS (Outcoming Seazures)

L'ensemble de la base de l'état nominal est composé de quatre semaines de simulations reparties en deux semaines pour l'apprentissage et deux semaines pour la validation.
Nous avons généré deux semaines pour l'état anormal, avec des perturbations de différents types, ces perturbations étant déclenchées à des temps choisis aléatoirement.
Les différentes perturbations utilisées sont :

- Surcharge globale
- Surcharge destination
- Surcharge origine
- Incident flux 200%

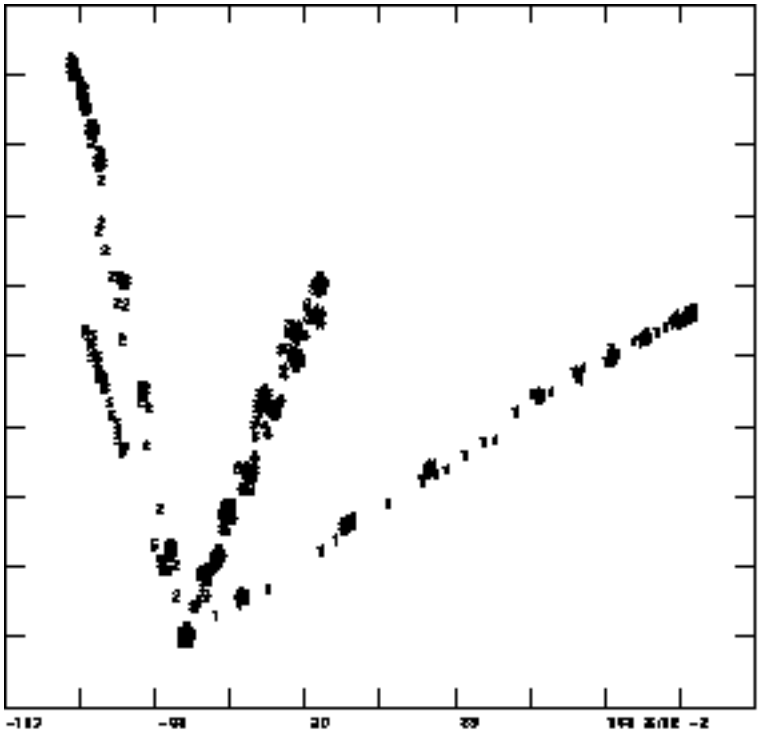

FIG. 9.11: *ACP sur l'ensemble des variables.*

Ces six semaines de données seront considérées comme six séries temporelles échantillonnées toutes les quatre minutes.

On trouve sur la figure 9.12 la série temporelle correspondante aux données relatives à la base d'apprentissage. Cette série montre de façon claire l'identification que l'on peut faire aisément entre le samedi et dimanche par rapport aux autres jours de la semaine (sur cette courbe ceux-ci sont identifiés par les pics les moins importants).

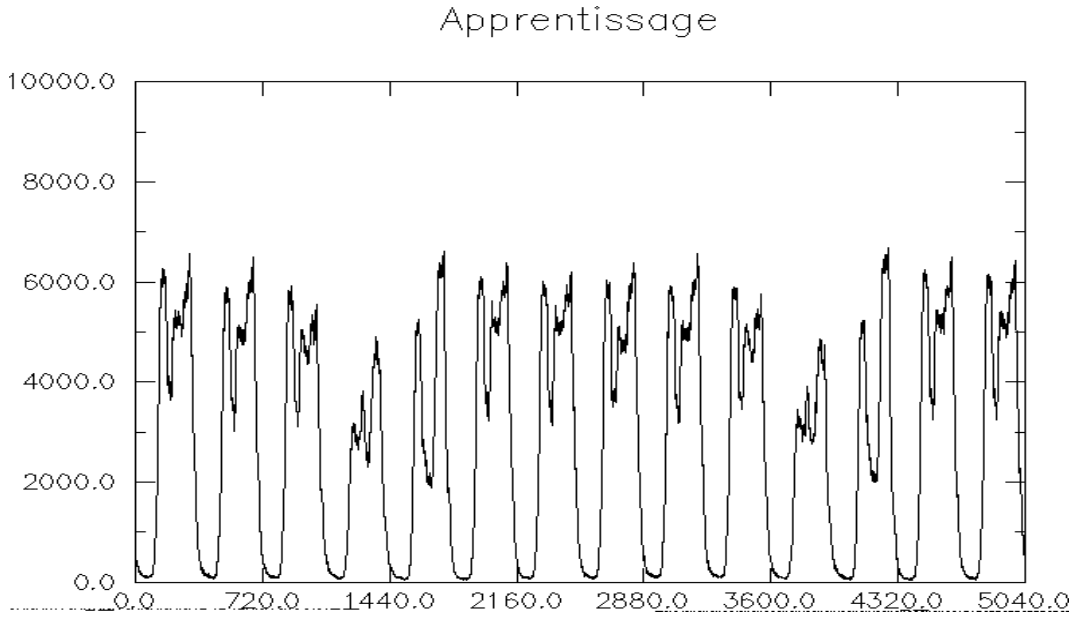

FIG. 9.12: *Série temporelle pour l'apprentissage.*

La figure 9.13 correspond à un échantillon de la base test.
Notons que les surcharges générées pour les différentes se traduisent par des fluctuations plus ou moins importantes de la série temporelle.

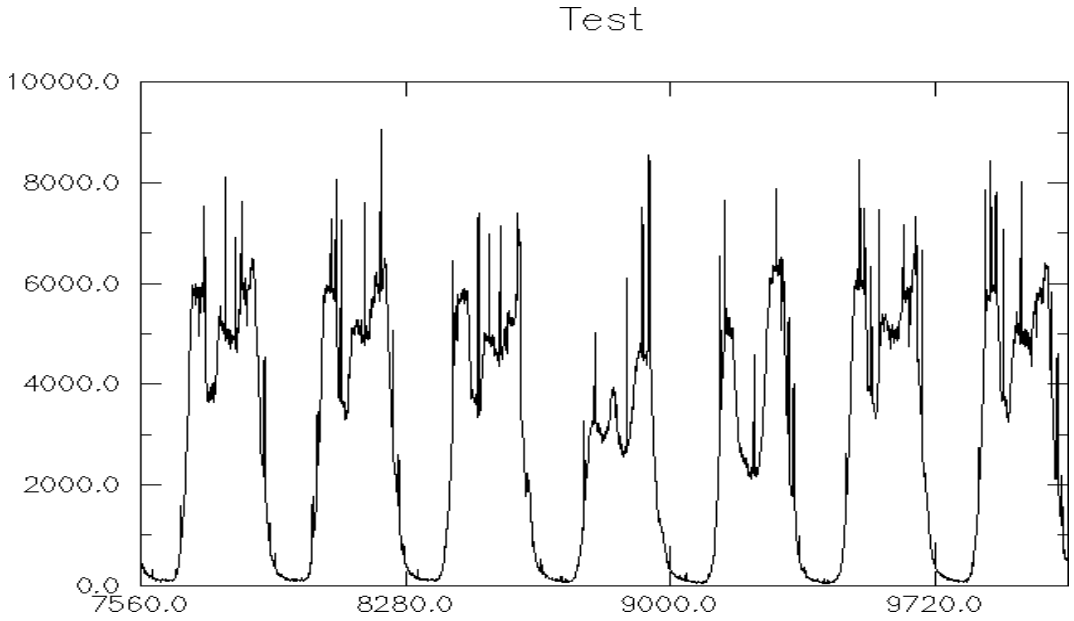

FIG. 9.13: *Série temporelle pour le test.*

Pour déterminer les entrées pertinentes de notre système de prédiction nous nous sommes basés sur l'étude du corrélogramme (Cf. Fig. 9.14) basée sur l'auto

corrélation de deux mesures prisent des temps différents.
En effet déterminer d'éventuelles corrélations, périodicites au niveau de la série, se trouve être essentiel pour tout système prédictif, hors l'élaboration d'un corrélogramme est un moyen simple à mettre en oeuvre pour une telle étude.

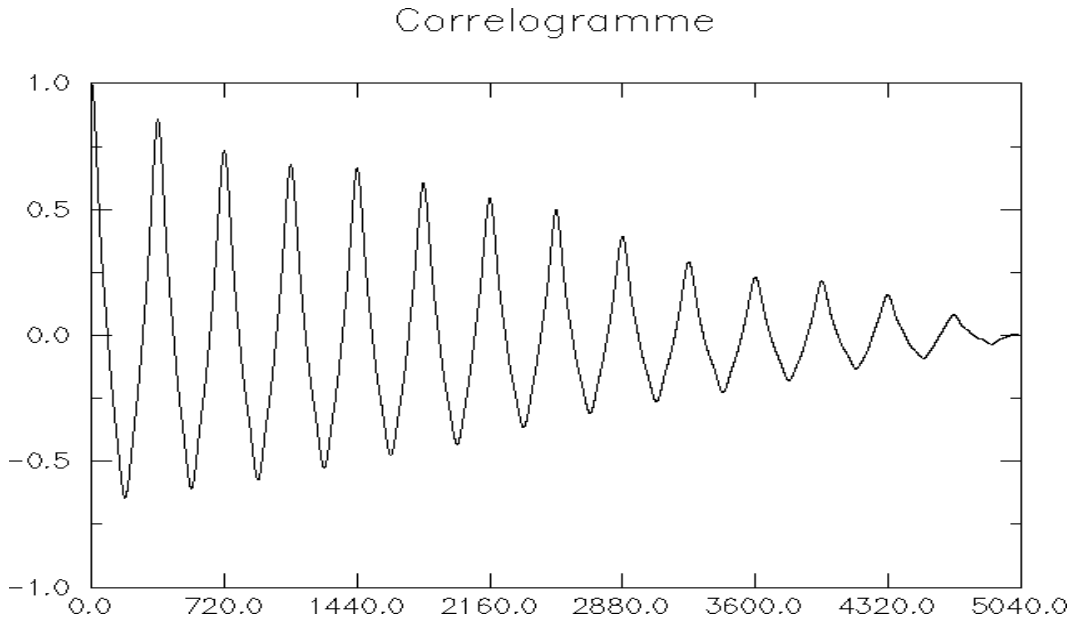

FIG. 9.14: *Corrélogramme de la série temporelle pour l'apprentissage.*

Le corrélogramme de la figure 9.14 montre une forte corrélation entre les mesures aux temps t-1 et t-2, ceux-ci seront donc intégrés dans l'entrée de notre système.
L'étude d'un corrélogramme effectué sur plusieurs semaines a mis en évidence une corrélation importante entre les instants t et t-2520, ceci s'explique par le fait que la mesure au temps t-2520 correspond à la mesure au temps t, si celle-ci avait été effectuée la semaine précédente.
Ces informations nous ont permis de déterminer l'entrée de notre système prédictif à savoir l'utilisation des mesures au temps t-1, t-2 et t-2520, il faut noter que l'ajout de la mesure au temps t-2520 apporte une information sur la valeur numérique à prédire au niveau quantitatif, mais intègre de façon implicite le jour et l'heure de la prédiction.

Bibliographie

[Bennani1992] BENNANI (Y.). – Approche connexionnistes pour la Reconnaissance Automatique du Locuteur : Modèlisation et Identification. *Thèse à l'université de Paris-Sud centre d'Orsay*, 1992.

[Bennani1993] BENNANI (Y.). – Probabilistic Cooperation Of Connectionist Expert Modules: Validation On A Speaker Identification Task. *IEEE , ICASSP'93, Minneapolis, Minnesota, U.S.A.*, 1993.

[Bennani1995] BENNANI (Y.). – Modular and Hybrid Connectionist System for Automatic Speaker Identification. *Neural Computation, MIT Press*, vol. 7(4), 1995, pp. 791–797.

[Bennani1998] BENNANI (Y.). – Contributions au Contrôle de la Capacité de Généralisation des Systèmes d'Apprentissage Connexionnistes. *Thèse d'Habilitation à Diriger des Recherches à l'université de Paris-Nord*, 1998.

[Bishop1994] BISHOP (C.). – Mixture Density Networks, Neural Computing Research Group Report. *NCRG/4288, Dept. of Comp. Sc., Aston University, Birmingham, UK*, 1994.

[Bochereau et Bourgine1990] BOCHEREAU (L.) et BOURGINE (P.). – Extraction of semantic features and logical rules from a multilayer neural network. *IJCNN*, vol. 2, 1990, pp. 579–582.

[Bollivier et al. 1991] BOLLIVIER (M. De), GALLINARI (P.) et THIRIA (S.). – Cooperation of Neural Nets and Task Decomposition. *IJCNN'91*, vol. 2, 1991, pp. 573–576.

[Bossaert1993] BOSSAERT (F.). – Rapport de D.E.A. *Université paris XIII, Laboratoire d'Informatique de Paris-Nord (LIPN)*, 1993.

[Boubour1997] BOUBOUR (R.). – Suivi de pannes par corrélation causale d'alarmes dans les systèmes répartis : Application aux réseaux de télécommunication. *Thèse à l'université de Rennes 1, IRISA*, 1997.

[Boutleux et Dubuisson1995] BOUTLEUX (E.) et DUBUISSON (B.). – Détection et Suivi d'Évolutions de l'État d'un Système Complexe: Application au réseau Téléphonique Français. *Personal Communication*, 1995.

[Breiman et al. 1984] BREIMAN (L.), FREIDMAN (J.), OLSHEN (R.) et STONE (C.). – Classification and Regression Trees. *Wadsworth Int. Group*, 1984.

[Brezellec et Soldano1993] BRÉZELLEC (P.) et SOLDANO (H.). – ELENA : A Bottom-Up learning method. *ICNL'93*, 1993, pp. 9–16.

[Cibas et al. 1994] CIBAS (T.), FOGELMAN SOULIE (F.), GALLINARI (P.) et RAUDYS (S.). – Variable Selection with Optimal Cell Damage. *ICANN'94*, 1994.

[Dague et al. 1997] DAGUE (P.), GUERRIN (F.) et TRAVÉ-MASSUYÈS (L.). – *Le Raisonnement Qualitatif pour les sciences de l'ingenieur.* – HERMES, 1997.

[Dague1994] DAGUE (P.). – Model-based diagnosis of analog electronic circuits. *Annals of Mathematics and Artificial Intelligence,special issue on Model-based Diagnosis, J.C. Baltzer*, vol. 11(1-4), 1994, pp. 439–492.

[De bois1994] DE BOIS (L.). – Time Series Forcasting or Network Traffic Management. *European Cooperation on Network Traffic Management*, 1994.

[De bruin et al. 1988] DE BRUIN (A.), RINNOOY KAN (A.H.G.) et TRIENEKENS (H.W.J.M.). – A simulation Tool for the performance Evaluation of Parallel Branch and Bound Algorithms. *Mathematical Programming*, vol. 42, 1988, pp. 245–271.

[Denoeux] DENOEUX (T.). – Génération automaitique de règles de classification par l'algorithme de rétropropagation du gradient. *AFCET AFIA, $3^{ème}$ journées du colloque symbolique numérique, Univ Paris IX.*

[Derijver et Kittler1982] DERIJVER (P.A.) et KITTLER (J.). – Pattern Recognition: a statitical approach. *Prentice-Hall International, London*, 1982.

[Didelet1992] DIDELET (E.). – Les arbres de neurones avec rejet d'ambiguité. Application au diagnostic pour le pilotage en temps réel du réseau téléphonique français. *Thèse de doctorat, Université Technologique de Compiègne*, 1992.

[Didelet1994] DIDELET (E.). – A Neural Technique Approach to Network Traffic Management. *ITC 14, Antibes, France*, 1994.

[Domart et Bourneuf1981] DOMART (A.) et BOURNEUF (J.). – Nouveau Larousse Medical. *Librairie Larousse*, 1981.

[Drucker et al. 1993] DRUCKER (H.), SCHAPIRE (R.) et SIMARD (P.). – Boosting Performance in Neural Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7(4), 1993, pp. 705–719.

[Dubuisson1990] DUBUISSON (B.). – Diagnostic et reconnaissance des formes. *Traité des Nouvelles Technologies, série Diagnostic et Maintenance, Hermes*, 1990.

[Fukunaga1990] FUKUNAGA (K.). – Statistical Pattern Recognition. *Academic Press*, vol. 2, 1990.

[Gallant1988] GALLANT (S.I.). – Connectionist expert systems. *ACM'88*, vol. 31, 1988, pp. 152–169.

[Ghosh et al. 1996] GHOSH (J.), BECK (S.) et DEUSER (L.). – Integration of Neural Classifiers for Passive Sonar Signals. *C.T. Leondes, editor, DSP Theory and Applications, Academic Press.*, 1996.

[Greiner et al. 1989] GREINER (R.), SMITH (B. A.) et WILKERSON (R. W.). – A correction to the algoithme in Reiter's theory of diagnosis. *Artificial Intelligence*, vol. 41(1), 1989, pp. 79–88.

[Guermeur et Gallinari1996] GUERMEUR (Y.) et GALLINARI (P.). – Combining Statistical Models for Protein Secondary Structure Prediction. *ICANN'96*, 1996.

[Hampshire et Waibel1992] HAMPSHIRE (J.B.) et WAIBEL (A.H.). – The Meta-Pi Network: Building Distributed Representations for Robust Multisource Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(7), 1992, pp. 751–769.

[Hansen et Salamon1990] HANSEN (L.K.) et SALAMON (P.). – Neural Network Ensembles. *Transactions on Pattern Analysis and Machine Intelligence*, 1990, pp. 993–1001.

[Hashem1992] HASHEM (S.). – Sensitivity Analysis for Feedforwrd Artificial Neural Networks with Differentiable Activation Functions. *International Joint Conference on Neural Networks, IJCNN'92*, vol. 1, 1992, pp. 419–424.

[Herrmann et al. 1989] HERRMANN (F.), STERN (D.) et CHEMOUIL (P.). – SUPERMAC: A Software Tool for the Performance Evaluation of Network Traffic Management. *ICCC, Symp. Beijing, China*, 1989.

[Hertz et al. 1991] HERTZ (J.), KROGH (A.) et PALMER (R.G.). – Introduction to the Theory of Neural Computation. *Addison Wesley*, vol. 1, 1991.

[Jacobs et al. 1991] JACOBS (R.A.), JORDAN (M.I.), NOWLAN (S.J.) et HINTON (G.E.). – Adaptive Mixtures of Local Experts. *Neural Computation*, vol. 3, 1991, pp. 79–87.

[Jobson1991] JOBSON (J.D.). – Applied multivariate data analysis. *Regression and experimental design, Springer-Verlag*, vol. 1, 1991.

[Konte et al. 1990] KONTÉ (A.), VICTORRI (B.) et RAYSZ (J.P.). – Rule extraction in recurrent connectionist networks. *Neuro-Nimes'90*, 1990, pp. 131–144.

[Lamy et Fogelman1995] LAMY (B.) et FOGELMAN (F.). – Can Multiple Initializations Improve Generalization? *ICANN'95*, vol. 1, 1995, pp. 39–44.

[Lamy1995] LAMY (B.). – Reconnaissance de caractères manuscrits par combinaison de modèles connexionnistes. *Thèse de doctorat l'université de Paris-6*, 1995.

[Le cun et al. 1990] LE CUN (Y.), DENKER (J.S.) et SOLLA (S.A.). – Optimal Brain Damage. *Neural Information Processing Systems*, vol. 2, 1990, pp. 598–605.

[Leray1998] LERAY (P.). – Early Brain Damge. *Thèse d'informatique de l'université paris 6 (LIP6)*, 1998.

[Linde et al. 1980] LINDE (Y.), BUZO (A.) et GRAY (R.M.). – An Algorithm for the VQ Design. *IIIE, Trans. on Communication*, vol. 28, 1980, pp. 84–95.

[Macqueen1967] MACQUEEN (J.). – Some Methods for Classification and Analysis of Multivariate Observations. *Fifth Berkeley Symposium on Mathematics, Statistics and Probabilities*, vol. 1, 1967, pp. 281–297.

[Mellouk et Gallinari1993] MELLOUK (A.) et GALLINARI (P.). – A discriminative neural prediction system for speech recognition. *ICASSP'93, Minneapolis, USA.*, 1993.

[Moody et Utans1992] MOODY (J.) et UTANS (J.). – Principed Architecture Selection for Neural Networks: Application to Corporate Bond Rating Prediction. *Neural Information Processing Systems*, vol. 4, 1992.

[Moody1994] MOODY (J.). – Prediction Risk and Architecture Selection for Neural Networks. *Statistics to Neural Networks-Theory and Pattern Rocgnition Application, Eds. V. Cherkassky, J.H. Friedmann, H.Wechsler, Springer-Verlag*, 1994.

[Nix et Weigend1995] NIX (D.A.) et WEIGEND (A.S.). – Learning Local Error Bars for Nonlinear Regression. *Advances in Neural Information Processing Systems, NIPS7, MIT Press*, 1995, pp. 489–496.

[Poggio et Girosi1990] POGGIO (T.) et GIROSI (F.). – Regularization algorithms that one equivalent to multilayer networks. *Science*, vol. 247, 1990, pp. 978–982.

[Poquelin1666] POQUELIN (J-B.). – Le Médecin malgré lui. *Nouveaux classiques Larousse, Librairie Larousse 1971*, vol. Acte II, scéne IV, 1666, p. 57.

[Raczkiewicz et Stern1993] RACZKIEWICZ (M.) et STERN (D.). – Methods to Detect Traffic Disturbances or Real-Time Network Management. *Technical Repport, DE/ATR/82-93*, 1993.

[Reiter1987] REITER (R.). – A theory of diagnosis from first principales. *Artificial Intelligence*, vol. 32(1), 1987, pp. 57–96.

[Ruck et al. 1990] RUCK (D.W.), ROGERS (S.K.) et KABRISKY (M.). – Feature selection using a multylayer perceptron. *Neural Network Comput*, vol. 2(2), 1990, pp. 40–48.

[Rumelhart et al. 1986] RUMELHART (D.E.), HINTON (G.E.) et WILLIAMS (R.J.). – Learning Internal Representations by Error Propagation. *Parallel Distributed Processing, MIT Press*, vol. 1, 1986.

[Saporta1978] SAPORTA (G.). – Théorie et Méthodes de la Statistique. *Éditions Technip*, 1978.

[Shavlik et Towell1991] SHAVLIK (J.W.) et TOWELL (G.G.). – The extraction of refined rules from knowledge-based neural networks. *Machine Learning'91*, 1991.

[Stern et Chemouil1992] STERN (D.) et CHEMOUIL (P.). – A Diagnosis Expert System or Network Traffic Management. *Networks92, Kobe, Japan*, 1992.

[Stern1991] STERN (D.). – A Statistical Study o Real-Time Telephone Traffic Variations or Network Management. *ITC Specialist Seminar, Krakw, Poland*, 1991.

[Stern1994] STERN (D.). – Supermac V3, 08 1994. CNET DE/ATR/08/94.

[Thiria et al. 1992] THIRIA (S.), MEJIA (C.), BADRAN (F.) et CREPON (M.). – Multimodular Architecture for Remote Sensing Operations. *in lippmann R., Moody J.E., Touretzky (ed.) Neural Information Processing System*, vol. 4, 1992.

[Tresp et al. 1997] TRESP (V.), NEUNEIER (R.) et ZIMMERMANN (G.). – Early Brain Damge. *Neural Information Processing Systems*, vol. 9, 1997, pp. 669–675.

[Tumer et Ghosh1996] TUMER (K.) et GHOSH (J.). – Theoretical Foundations of Linear and Order Statistics Combiners for Neural Pattern Classifiers. *TR-95-02-98, CVRC, University of Texas, Austin*, 1996.

[Turner et Gedeon] TURNER (H.S.) et GEDEON (T.D.). – Extracting meaning from neural networks. $11^{ème}$ *journées d'intelligence artificielle d'Avignon.*

[Wolpert1992] WOLPERT (D.H.). – Stacked Generalization. *Neural Computation, MIT Press*, vol. 5(2), 1992, pp. 241–259.

[Xu et al. 1992] XU (L.), KRZYZAK (A.) et SUEN (C.Y.). – Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *Transactions on Systems, Man and Cybernetics*, vol. 22(3), 1992, pp. 418–435.

[Yacoub et Bennani1997] YACOUB (M.) et BENNANI (Y.). – HVS : A heuristic for variable selection in multilayer artificial neural network classifier. *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 7, 1997, pp. 527–532.

Titre : Approches connexionnistes pour le diagnostic des systèmes complexes: application au réseau téléphonique

Résumé:
Dans cette thèse, nous nous intéressons à l'utilisation et l'adaptation des techniques connexionnistes dans la réalisation des systèmes de diagnostic de systèmes complexes. Nous abordons le problème difficile de la sélection de variables et nous proposons deux nouvelles mesures de pertinence permettant de quantifier l'importance de chaque variable dans le système d'apprentissage. Ces techniques d'élagage permettent d'une part, d'ajuster la complexité du modèle à la difficulté du problème et d'autre part, de sélectionner un sous-ensemble de caractéristiques pertinentes. Nous abordons ensuite le problème du diagnostic des systèmes complexes en utilisant les techniques de sélection de variables comme pré-traitement des données. Nous proposons plusieurs systèmes connexionnistes multi-modulaires pour réaliser d'une part, des tâches de diagnostic sur le réseau téléphonique visant à assurer la détection d'incidents et d'autre part, l'identification de perturbations. Nous étudions ces deux problèmes du diagnostic (détection et identification) avec deux approches différentes : l'approche par modélisation connexionniste et l'approche par combinaison de modèles.

Title : Connectionist approaches for complex systems diagnosis : application to telephonic network

Abstract:
In this thesis, we are interested in the connectionist technique utilization and adaptation in the realization of diagnosis systems of complex systems. First, we deal with the difficult problem concerning variable selection, and we propose two new pertinence measures allowing variable importance quatification in the learning system. These pruning technics permit on the one hand, to fit the model complexity with on the order hand the problem diffculty, to select an underset of pertinent characteristics. Then we deal with the complex system diagnosis problem using variable selection technics as a data preprocessing. We propose various multi-modular connectionist systems to realize, on the one hand, diagnosis works on the telephonic network in order to assure incident detection and on the other hand, identification of perturbations. We deal with these two diagnosis problems (detection and identification) with two different approaches: the connectionist modelling approach, and the model combination approach.