UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE Mathématiques, Informatique, Physique Théorique et Ingénierie des Systèmes (MIPTIS)

Laboratoire PRISME

THÈSE présentée par :

Nesrine Keltoum KHODJA

Soutenue le : **23 octobre 2024**

Pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline/ Spécialité : **Sciences et technologies industrielles**

Contribution à l'apprentissage des Modèles de Markov Cachés pour l'estimation de l'état de santé de systèmes industriels

THÈSE dirigée par :

M. Manuel AVILA	Maître de Conférences, HDR, Université d'Orléans

RAPPORTEURS :

Mme Zineb SIMEU-ABAZI	Professeur des Universités, INP Grenoble
M. Nacim BOUDAOUD	Maître de Conférences, HDR, UTC COMPIEGNE

JURY :

Mme Anne BARROS	Professeur des Universités, Université Paris Saclay (Présidente)
M. Fréderic KRATZ	Professeur des Universités, INSA CVL
M. Thierry PAQUET	Professeur des Universités, Université de Rouen
Mme Zineb SIMEU-ABAZI	Professeur des Universités, INP Grenoble
M. Nacim BOUDAOUD	Maître de Conférences, HDR, UTC COMPIEGNE
M. Manuel AVILA	Maître de Conférences, HDR, Université d'Orléans
M. Stéphane BEGOT	Maître de Conférences, Université d'Orléans
M. Florent DUCULTY	Maître de Conférences, Université d'Orléans

INVITÉS :

M. Mohamed SLIMANE Professeur émérite, Université de Tours

Nesrine Keltoum KHODJA

Contribution à l'apprentissage de Modèles de Markov Cachés pour l'estimation de l'état de santé de systèmes industriels

Résumé :

Une bonne politique de maintenance au sein d'une entreprise est un enjeu majeur pour assurer la fiabilité et la sûreté de fonctionnement des outils de production dans différents secteurs.

Le travail de recherche présenté dans cette thèse vise à introduire de nouvelles méthodes de diagnostic des défauts des systèmes industriels. Ces défauts sont vus comme une problématique de multi-classe. C'est pour cette raison que nous avons choisi les méthodes orientées données fondées sur la classification et l'identification.

Nous avons constaté à travers les recherches bibliographiques que les Modèles de Markov Cachés (MMC) ont la capacité à modéliser les systèmes complexes. Nous avons utilisé ces modèles pour apprendre la signature particulière annonçant l'arrivée d'une panne.

Dans cette thèse, nous allons nous focaliser sur le cas de la mise en compétition de plusieurs modèles qui representent plusieurs modes de défaillance et un mode sans défaut. Dans ces travaux de recherche, nous souhaitons améliorer l'apprentissage des MMC. Nous avons développé une nouvelle stratégie d'apprentissage. Cette stratégie s'inspire des principes de non occurrence, c'est à dire, apprendre le contraire d'une forme. Pour améliorer la phase d'apprentissage des MMC multi-classes, il est envisagé de considérer les 'non-classes'. Les 'Non-modèles' seront appris en considérant uniquement les échantillons des autres classes.

Deux problématiques ont été posées : Est-ce-que les Non-modèles ont la capacité de reconnaitre les classes ? Est-ce-que la combinaison des Modèles et des Non-modèles améliore l'apprentissage des Modèles de Markov Cachés ? Nous répondons à ces deux problématiques à travers nos protocoles de test illustrés dans cette thèse. Nous expliquons, ensuite, le nouvel algorithme que nous avons développé qui utilise à la fois les informations contenues dans les Modèles et les Non-modèles. Finalement, nous discutons des résultats obtenus et nous les comparons aux usuels algorithmes d'apprentissage de MMC.

Mots clés : Modèles de Markov Cachés, apprentissage, classification.

Contribution to the learning of Hidden Markov Models for estimating the state of health of industrial systems

Abstract :

A good maintenance policy within a company is a major factor in ensuring the reliability and dependability of production tools in various sectors (aeronautics, military, nuclear, etc.). The aim of the research work presented in this thesis is to introduce new methods for diagnosing faults in industrial systems. These faults are seen as multi-class problem, which is why we have chosen data-driven methods based on classification and identification. We have found through literature searches that Hidden Markov Models (HMM) have the ability to model complex systems. To this end, we have used these models to learn the particular signature announcing the imminent arrival of a fault. In this thesis, we will focus on the case of competing models representing several failure modes and a fault-free mode. In this thesis, we aim to improve HMM learning, so we have developed a new learning strategy. This strategy is inspired by the principles of non-occurrence, i.e. learning the opposite of a shape. To improve the learning phase of multi-class HMM problems, we plan to consider 'non-classes'. 'Non-models' will be learned by considering only samples from other classes. Two issues were raised : Do the Non-models have the ability to recognize classes ? Does the combination of Models and Non-models improve Hidden Markov Model learning ? We answer these two questions through our protocol and tests illustrated in this thesis, as well as explaining the new algorithm we've developed that uses both the information contained in the Models and the information contained in the Non-models. Finally, we discuss the results obtained and compare existing HMM learning algorithms with our new algorithm.

Keywords : Hidden Markov Models, training, classification

Laboratoire PRISME, Université d'Orléans

8 Rue Léonard de Vinci - 45072 Orléans

Remerciements

Les travaux que je présente dans ce manuscrit n'auraient pas été possibles sans l'aide de nombreuses personnes dont : mon directeur de thèse, Manuel AVILA, pour m'avoir accueilli dans son équipe et m'avoir fait confiance depuis mon arrivée en France. J'ai ainsi découvert le monde de la recherche au sein de l'équipe IRAuS du laboratoire PRISME.

J'adresse toute ma reconnaissance à mes co-encadrants Florent DUCULTY (pour avoir corrigé mes fautes d'ortographe) et à Stéphane BEGOT (pour la corrections à « quatre mains » que nous avons réalisée durant de longues heures). Tous deux sont enseignants chercheurs à l'institut PRISME.

J'exprime mes sincères remerciements à Monsieur Nacim Boudaoud et Madame Zineb SIMEU-ABAZI pour l'intérêt qu'ils ont bien voulu me porter en acceptant le rôle de rapporteur dans mon jury de thèse.

J'exprime mes sincères remerciements à Messieurs Frédéric KRATZ, Thierry PAQUET ainsi qu'à Madame Anne BARROS qui m'ont également fait l'honneur de participer au jury ; enfin, je remercie Monsieur Mohamed SLIMANE d'avoir accepté l'invitation à ma soutenance.

J'adresse aussi à toutes les valeureuses personnes présentes au sein du laboratoire qui ont croisé ma route, de sincères remerciements. J'ai apprécié l'excellente cohésion présente entre les équipes de recherche, et j'espère y avoir contribué. Je tiens aussi à remercier tout le personnel de l'IUT de l'Indre qui m'a encouragée pendant toutes ces années.

Enfin, je remercie mes parents de m'avoir encouragée à faire de la recherche jusqu'à voir leur fille devenir docteur.
Je remercie aussi mes deux soeurs Zahra et Sabrine, mon beau-frère Hichem pour son aide, ainsi que la famille Favière (Raphaël, Jean-Marie, Soraya, Florian et Ana) pour leur soutien.

Sommaire

Liste des figures v

Liste des tables ix

1 Chapitre 1 : Etat de l'art 9

1.1 Introduction . . . 9

1.2 La maintenance . . . 10

1.3 Objectifs de la maintenance . . . 12

1.4 Méthodes de maintenance . . . 13

1.4.1 La maintenance corrective . . . 14

1.4.2 La maintenance préventive . . . 14

1.5 Optimisation de la maintenance . . . 18

1.6 Concepts de diagnostic et de pronostic pour la maintenance . . . 19

1.6.1 Le diagnostic . . . 19

1.7 Analyses comparatives des méthodes de diagnostic . . . 25

1.7.1 Le pronostic . . . 28

1.8 Conclusion . . . 30

2 Chapitre 2 : Les Modèles de Markov Cachés 33

2.1 Introduction . . . 33

2.2 Les Modèles de Markov Cachés (MMC) . . . 33

2.2.1 Définition des MMC . . . 35

2.2.2 Présentation des paramètres des MMC . . . 36

2.2.3 Principales architectures des MMC . . . 38

2.2.4 Trois problèmes fondamentaux pour les MMC . . . 38
2.2.5 Les algorithmes de base . . . 39
2.2.6 Apprentissage des MMC . . . 43
2.2.7 Comparaison entre Baum-Welch et Viterbi . . . 48
2.3 Intérêt des Modèles de Markov Cachés . . . 48
2.4 Limitation des MMC . . . 50
2.5 Problèmatique multi-classe . . . 50
2.6 Conclusion . . . 52

3 Chapitre 3 : Méthode de classification par les Modèles de Markov Cachés 55
3.1 Introduction . . . 55
3.2 Modèles de synthèse . . . 56
3.2.1 Un modèle de synthèse par classe . . . 57
3.2.2 Répartition des données en corpus d'apprentissage et de test . . . 60
3.3 Classification par Modèles de Markov Cachés . . . 62
3.4 Méthode proposée . . . 63
3.4.1 Phase d'apprentissage . . . 64
3.4.2 Phase de test . . . 65
3.4.3 Résultats de classification . . . 65
3.5 Confiance dans les réponses de classification . . . 67
3.6 Vérification de la pertinence des données . . . 72
3.6.1 Mesure d'entropie . . . 72
3.6.2 Information mutuelle . . . 73
3.6.3 Entropie et information mutuelle vs taux de reconnaissance . . . 74
3.6.4 Évolution de l'entropie selon la taille du corpus de données . . . 76
3.7 Conclusion . . . 77

4 Chapitre 4 : Modèles Vs Non-Modèles 79
4.1 Introduction . . . 79
4.2 Définition du terme Non-modèle . . . 80
4.3 Discrimination par Non-modèle . . . 82
4.3.1 Formation des corpus d'apprentissage des Non-modèles . . . 82

4.3.2 Apprentissage des MMC par les Non-modèles 84
4.3.3 Phase de Test 85
4.4 Résultats de classification 85
4.5 Confiance dans la réponse de classification 88
4.5.1 Discussion sur les résultats obtenus 93
4.6 Algorithme d'apprentissage Non-Behavior Viterbi (NB Viterbi) 94
4.6.1 Explication de l'algorithme NB Viterbi 95
4.7 Résultats obtenus par NB Viterbi 98
4.7.1 Comparaison des résultats obtenus par apprentissage NB Viterbi avec les autres algorithmes d'apprentissage 99
4.7.2 impact des corpus Non-Modèle sur les résultats obtenus 100
4.8 Variation du nombre d'états des MMC 101
4.9 Conclusion 102

Bibliographie **111**

Liste des figures

1 Un incendie dans une usine contenant des produits chimiques au sud-ouest de Chicago [1] . . . 3

1.1 Composantes de la surveillance industrielle . . . 12
1.2 Les types de maintenance . . . 13
1.3 Le choix de la bonne politique de maintenance . . . 18
1.4 Générateur de résidus à base d'un observateur . . . 22
1.5 Générateur de résidus dans l'espace de parité . . . 22
1.6 Description schématique d'une estimation paramétrique . . . 23
1.7 La différence entre le diagnostic et le pronostic . . . 28
1.8 Différentes méthodes de pronostic . . . 29

2.1 Visualisation réseau . . . 34
2.2 Evolution des travaux qui utilisent les MMC pour le pronostic et le diagnostic . . . 35
2.3 Relations de dépendance entre les variables aléatoires d'un MMC . . . 37
2.4 Différentes topologies de MMC ((a) ergodique, (b) gauche-droite) . . . 38
2.5 Exemple de deux chemins les plus probables de Viterbi . . . 47
2.6 Problème des multi-classes . . . 51
2.7 Principe de comportement et non-comportement . . . 52

3.1 Exemple avec 4 classes (3 modes de défaillance plus un mode sans défaillance) . . . 56
3.2 Topologie de type gauche-droite . . . 58
3.3 Production des bases de données . . . 59
3.4 Distribution des symboles pour les corpus apprentissage et test de la classe 1 . . . 60
3.5 Distribution des symboles pour les corpus apprentissage et test de la classe 2 . . . 61

3.6 Distribution des symboles pour les corpus apprentissage et test de la classe 3 . . 61
3.7 Distribution des symboles pour les corpus apprentissage et test de la classe 4 . . 62
3.8 Les étapes de classification par MMC avec données de synthèse 64
3.9 Résultat de classification en utilisant Baum-Welch et Viterbi 66
3.10 Ecart entre la classe 1 et les autres classes 68
3.11 Ecart entre la classe 2 et les autres classes 69
3.12 Ecart entre la classe 3 et les autres classes 70
3.13 Ecart entre la classe 4 et les autres classes 71
3.14 Résultats d'entropie pour dix tirages pour les corpus d'apprentissage et de test . 73
3.15 Information mutuelle des dix tirages pour les corpus d'apprentissage et de test . . 74
3.16 Comparaison entre l'entropie et le taux de reconnaissance avec apprentissage Baum Welch 75
3.17 Comparaison entre information mutuelle et taux de reconnaissance avec apprentissage Baum-Welch 75
3.18 Evolution de l'Entropie de Shannon selon la taille du corpus de données 76

4.1 Principe des Modèles et Non-modèles 80
4.2 Formation des corpus d'apprentissage Non-modèle 82
4.3 Ensemble d'apprentissage du Non-modèle 1 83
4.4 Ensemble d'apprentissage du Non-modèle 2 83
4.5 Ensemble d'apprentissage du Non-modèle 3 84
4.6 Ensemble d'apprentissage du Non-modèle 4 84
4.7 Méthode d'identification fondée sur les non-modèles 85
4.8 Résultat de classification Modèle et Non-modèle avec Baum-Welch 86
4.9 Résultats de classification des Modèles et des Non-modèles avec Viterbi 87
4.10 Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 1 89
4.11 Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 2 90
4.12 Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 3 91

4.13 Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 4 92
4.14 Illustration des meilleurs chemins de Viterbi pour les Non-modèles pour deux séquences 96
4.15 Résultat de classification sur dix tirages aléatoires 98
4.16 Résultats obtenus par les trois algorithmes d'apprentissage sur dix tirage 99
4.17 Variation du poids des Non-modèles 101
4.18 Variation du nombre d'états avec apprentissage Viterbi et NB Viterbi 101
4.19 différence des taux de reconnaissance en pourcentage entre Viterbi et NB Viterbi 102

Liste des tableaux

1.1 Avantages et inconvénients des maintenances corrective et préventive 17

1.2 Comparaison des différentes méthodes utilisées pour le diagnostic des défauts . . 27

2.1 Contribution des deux chemins de Viterbi 47

3.1 Matrice de confusion du premier tirage avec apprentissage Baum Welch 66

3.2 Matrice de confusion du premier tirage avec apprentissage Viterbi 67

4.1 Matrice de confusion du premier tirage avec apprentissage Baum Welch 87

4.2 Matrice de confusion du premier tirage avec apprentissage Baum Welch 88

4.3 Extrait des séquences de la base de test identifiées par les classes et les Non-classes avec apprentissage Viterbi 93

4.4 Contribution des deux chemins de Viterbi 96

4.5 Matrice de confusion du premier tirage avec apprentissage NB Viterbi 99

Glossaires

S=$\{S_1, S_2, ..., S_N\}$: L'ensemble des différents états.

$V = \{V_1, V_2, ..., V_M\}$: L'ensemble discret de symboles d'observation possibles.

$Q = \{q_1, q_2, ..., q_T\}$: La séquence d'états avec T la longueur de la séquence, l'état à l'instant t est q_t.

$\{O\} = \{O_1, O_2, ..., O_T\}$: La séquence d'observations où chaque observation O_t est l'un des symboles de V.

A = $\{a_{ij}\}$: La distribution de probabilité de transition d'état

B=$\{b_j(\text{k})\}$: La distribution de probabilité des symboles d'observation dans l'état j.

π : La distribution d'état initial.

λ : Le Modèle de Makov Caché.

$\{O_{app}\}$: Séquence d'observations d'apprentissage.

$\{O_{test}\}$: Séquence d'observations à identifier (Test).

n_{ij} : Le nombre de fois ou la transition $i \rightarrow j$ est utilisée

n_i : Le nombre de fois ou l'état i est utilisé.

n_i^O : Le nombre de fois où le symbole O est généré par l'état i.

$\bar{\lambda}$: Le Modèle réestimé.

$\bar{\lambda}_{Nm}$: Le Non-modèle réestimé.

R : Le nombre de classes.

r : Le numéro de la classe.

M : Le nombre de symboles.

N : Le nombre d'états.

L_{tot} : La taille d'une base de données.

L : La taille du corpus d'apprentissage du Modèle.

$L_{tot} - L$: La taille du corpus de test.

P : La taille du corpus d'apprentissage du Non-modèle.

l : Le numéro de la séquence Modèle.

p : Le numéro de la séquence Non-modèle.

α_i : La variable forward.

β_i : La variable Backward.

$\{O_{app\ Nm}\}$: La séquence d'apprentissage des Non-modèles.

Liste des abréviations

MMC : Modèle de Markov Caché.

HMM : Hidden Markov Model.

PHM : Prognostics and Health Management.

ACP : l'analyse en composantes principale.

App : apprentissage.

PB : Comportement positif.

CNR : Comportement Non Récurrent.

NB : Non-Behavior.

DNN : Réseaux neuronaux profond.

Introduction Générale

L'industrie est sans cesse en mouvement. Elle se perfectionne, se délocalise, développe et invente dans le but de garder ou de gagner des parts de marché face à une concurrence féroce. Néanmoins, les activités industrielles et humaines font presque quotidiennement les grands titres des actualités avec les différents accidents ou d'évènements catastrophiques. En effet, le zéro défaut ou le risque zéro n'existe malheureusement pas pour les activités industrielles.

Les catastrophes industrielles, principalement les explosions d'usines chimiques ou les fuites de pétrole, causent d'énormes dégâts à la fois aux personnes et à l'environnement. Elles nous rappellent à quel point la sécurité dans les usines et les normes strictes sont vitales pour éviter de telles tragédies (voir la figure 1).



FIGURE 1 – Un incendie dans une usine contenant des produits chimiques au sud-ouest de Chicago [1]

Toutefois, pour tenter de réduire les risques à un niveau le plus faible possible et acceptable, des

méthodes, des techniques et des outils scientifiques ont été développés dès le début du XXème siècle pour évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques lorsqu'elles se produisent. Les pannes des équipements industriels représentent un coût très important pour les entreprises et les outils de production. Dans le secteur manufacturier, la Société Internationale d'Automatisation estime que les industriels perdent chaque année 647 milliards de dollars dans le monde en raison des temps d'arrêt. Pour lutter contre cet état de fait, depuis le 1er février 2024, la Cour des comptes a publié une enquête à propos de la « gestion des risques liés aux installations classées pour la protection de l'environnement (ICPE) dans le domaine industriel ». Portant sur la période 2010-2022, l'enquête constate les progrès accomplis en matière de prévention des risques industriels tout en relevant de nombreux points d'amélioration [2].

Le diagnostic et le pronostic de pannes des systèmes industriels sont devenus une des priorités au sein de l'entreprise. Les méthodes utilisées pour le diagnostic de pannes ou de défauts ont largement évolué ces dernières années. Les chercheurs dans le domaine de la maintenance, ne cessent de développer des moyens d'anticipation, ce qui les a amenés à user de toutes les techniques, y compris celles de l'intelligence artificielle (IA), en vue de fiabiliser les diagnostics et voire même, de prédire les défauts et les pannes, afin de maitriser les temps d'arrêt, qualifiés souvent de temps morts. Plusieurs techniques de maintenance ont été developpées. Elles ont d'abord été utilisées dans la maintenance corrective afin de diagnostiquer des défauts apparents, puis dans la maintenance systématique pour au final être utilisées dans la maintenance préventive conditionnelle. La maintenance systématique est définie comme la réalisation d'une maintenance périodique visant à ce qu'un système ne tombe pas en panne. La maintenance préventive conditionnelle, quant-à-elle, est définie comme une action de maintenance prise en se fondant sur des informations acquises à partir de mesures effectuées sur le système. L'efficacité de cette dernière est jugée à partir des méthodes de diagnostic utilisées et qui sont en constante amélioration grâce aux recherches effectuées dans ce domaine. Ces recherches, initialement dédiées au diagnostic des défauts, se sont étendues ces dernières années à la prédiction des défauts (pronostic). Le diagnostic et le pronostic des défauts d'un système se révèlent être une stratégie clé et prometteuse qui suscite auprès des industriels énormément d'intérêt car elle permet d'augmenter la fiabilité et la sécurité des systèmes d'entraînement électriques mais aussi de réduire les coûts liés à leur maintenance. Il

existe deux grandes familles d'approches pour le diagnostic et le pronostic des défauts : l'approche fondée sur des modèles analytiques et l'approche fondée sur des méthodes orientées données. Le choix d'une approche est fondé sur la complexité du système, mais aussi sur la disponibilité de la connaissance que l'on souhaite acquérir sur lui. En effet, tous les systèmes sont susceptibles d'évoluer vers différents modes de fonctionnement. Ces modes ne sont pas obligatoirement tous connus car certains d'entre eux peuvent n'avoir été jamais observés sur le système. Pour résoudre ces problèmes, il convient de développer des méthodes de diagnostic disposant d'une certaine faculté d'adaptation à des situations nouvelles et permettant une reconnaissance fiable du mode de fonctionnement dans lequel le système évolue. Il convient aussi de développer des méthodes de pronostic capables d'estimer la durée de vie restante avant l'apparition d'un défaut et de détecter l'imminence d'un ou plusieurs défauts.

Dans ce manuscrit, nous considérons que le système industriel peut être soumis à différents modes de défaillance. Nous supposons que nous disposons d'un retour d'expérience pour ces défaillances sous forme d'événements enregistrés pour chaque type de défaillance ou pour l'absence de défaillance. Ces modes peuvent être considérés comme un problème multi-classes avec des échantillons enregistrés pour chaque classe. Pour cela, nous utilisons les méthodes de diagnostic orientées données. Nous employons la classification par les Modèles de Makov Cachés (MMC). En effet, la notion de Modèles de Markov Cachés, apparaît dans de nombreux domaines comme la théorie de l'information, la physique statistique, la compression de données, la bio-informatique (segmentation de gènes ou protéines afin de comparer des séquences d'ADN), la reconnaissance vocale, la maintenance industrielle, etc. Cette modélisation est une approche prometteuse dans certains domaines où l'on cherche à conceptualiser des données partiellement manquantes, comme la reconnaissance de caractères dans l'écriture manuscrite ou la reconnaissance d'empreintes digitales.

Dans cette thèse, nous nous sommes focalisés, en particulier, sur la mise en compétition de plusieurs MMC et nous souhaitons développer une nouvelle stratégie d'apprentissage des MMC qui consiste à apprendre le contraire d'une forme (non-classe). Pour cela deux problématiques ont été posées : est-ce-que les informations contenues dans les non-classes sont pertinentes et est-ce que ces informations sont complémentaires aux informations contenues dans les classes ?

Ce manuscrit comporte quatre chapitres. Le **premier chapitre** présente un état de l'art sur les méthodes existantes en matière de diagnostic et de pronostic utilisées dans la maintenance préventive conditionnelle. Les méthodes de diagnostic et de pronostic, mises au point ces dernières années, sont examinées et comparées. Il en ressort de cette comparaison que les méthodes de diagnostic et de pronostic orientées données permettent de modéliser le comportement d'un système (processus de dégradation, défaillance) uniquement à partir des données de mesure. Le diagnostic utilise la reconnaissance des formes pour détecter des changements dans les données de mesure via des méthodes de classification supervisée.

Dans **le second chapitre**, nous commençons par une étude bibliométrique et une visualisation réseaux à l'aide du logiciel VOSViewer pour visualiser la relation entre les MMC et le diagnostic ainsi que le pronostic. Par la suite, nous faisons une étude bibliographique sur les travaux effectués dans le contexte de la maintenance en utilisant les MMC. Ensuite, nous donnons la définition des Modèles de Markov Cachés. Nous citons les différents problèmes des MMC ainsi que les algorithmes de base et les differents algorithmes d'apprentissage. Enfin nous aborderons la problématique des multi-classes.

Le troisième chapitre est consacré à l'application de notre méthode. Nous allons expliquer comment nous avons géneré les différentes classes qui représentent les différents modes de fonctionnement. Ensuite, nous expliquerons les modèles de synthèse utilisés. Nous montrons comment nous avons réparti les corpus d'apprentissage et de test. Nous expliquerons aussi la méthode usuelle de classification par MMC anisi que les résultats obtenus en utilisant les différents algorithmes d'apprentissage. Ces résultats nous serviront de références pour la suite de cette thèse. Enfin, nous verifions la pertinence des données en utilisant les différentes théories de l'information.

Dans **le quatrième chapitre**, nous introduisons notre nouvelle méthode d'apprentissage qui consiste à apprendre le contraire d'une forme. Dans un premier temps, nous répondons à la première problématique qui consiste à verifier si les non-classes ont la capacité de descrminer les classes. Une fois cette problématique verifiée, nous expliquons notre nouvel algorithme qui consiste à utiliser l'information contenue dans les classes ainsi que l'information contenue dans les non-classes afin

d'améliorer l'apprentissage des classes.
Nous montrons que le nouvel algorithme apporte de meilleurs résultats. Nous faisons varier le nombre d'états des MMC pour verifier que le nouvel algorithme s'adapte à la compléxité des MMC.

Dans une **conclusion générale**, nous repositionnons l'ensemble de nos méthodes de diagnostic au regard de l'objectif initial de cette thèse « Contribution à l'apprentissage des Modèles de Markov Cachés pour l'estimation de l'état de santé des systèmes industriels ». Nous résumerons les principaux résultats de notre travail et nous aborderons les perspectives de travail qui découlent de cette thèse.

Chapitre 1

Etat de l'art

1.1 Introduction

Les systèmes autonomes ou partiellement autonomes, tels que les aéronefs, satellites, drones, rovers, voitures, humanoïdes, etc., sont devenus si complexes qu'il devient humainement difficile d'anticiper leurs comportements. Ainsi, l'exigence vis-à-vis de l'activité de maintenance dans les entreprises ne cesse d'augmenter de manière significative. La fiabilité et la sûreté de fonctionnement d'un système industriel, à moindre coût, sont les principaux objectifs des entreprises industrielles pour diminuer l'impact économique majeur dû aux pannes des systèmes, les pertes de production qui s'en suivent et la maintenance des installations. La maintenance contribue à garantir la continuité des opérations, à minimiser les coûts, à améliorer la sécurité, à maintenir la qualité et à assurer la conformité aux normes et réglementations. Elle est une partie intégrante de la gestion efficace des actifs et des opérations dans divers secteurs. Il est donc essentiel que la conception et l'utilisation du système s'élargissent à la gestion de la santé (System Health Management, Prognostics and Health Management ou PHM). La gestion de santé d'un système vise au développement d'outils qui aident à la réduction des coûts liés aux indisponibilités et aux réparations. Le présent chapitre est organisé comme suit. Dans la section 1.2, nous commençons par la présentation de notions élémentaires en maintenance des systèmes industriels. Dans la section 1.3, nous citons les objectifs de la maintenance. Dans la section 1.4, nous allons montrer les principaux aspects de la mise en œuvre d'une maintenance préventive. Nous allons décrire ensuite les bénéfices pouvant être attendus par la mise en œuvre d'une approche de diagnostic et de pronostic au service de la maintenance préventive conditionnelle. De manière plus ciblée,

ce chapitre donne une vue globale sur les méthodes existantes en matière de diagnostic et de pronostic. L'ensemble permet de montrer la diversité des approches envisageables et de choisir parmi elles celle qui définira un cadre de travail pour notre thèse.

1.2 La maintenance

La maintenance industrielle à travers les outils industriels et les équipements de production est un point de vigilance important pour les entreprises de tous les secteurs. En effet, une bonne politique de maintenance industrielle va permettre de réduire les coûts de production, d'améliorer la qualité ou d'anticiper les arrêts de production. À terme, la maintenance industrielle participe à l'amélioration du taux de productivité et/ou parfois à la réalisation d'économies sur sa chaîne de production. Aujourd'hui, le secteur de la maintenance industrielle représente l'équivalent de 22,6 milliards d'euros de dépense dans l'industrie [3].

Définition 1 : « Maintenance » : mot dérivé du verbe « maintenir », est issu des termes latins « manu » qui signifie main et « tenere » tenir. Il désigne « Ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise ». Par définition, la maintenance est donc fortement liée à la notion d'outil. La maintenance serait donc apparue avec les premiers outils. La dématérialisation des outils de travail au XXème siècle a provoqué une altération profonde de son champ d'application. A la suite de ces évolutions, ce périmètre englobe aujourd'hui notamment le maintien de processus, de programmes informatiques et, plus généralement, de systèmes d'information. Au fil du temps, la maintenance s'est déclinée en diverses stratégies qu'il convient de définir et catégoriser. A cette fin, ce manuscrit s'appuiera sur les concepts normalisés par la norme AFNOR NF EN 13306 [4]. Par conséquent, la maintenance sera à présent considérée comme l'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise. Un bien désigne ici tout « élément, composant, mécanisme, sous-système, unité fonctionnelle, équipement ou système qui peut être considéré individuellement ».

Définition 2 : « Défaillance ». Une défaillance correspond à une cessation de l'aptitude d'une entité à accomplir une ou plusieurs fonctions requises. On peut repérer deux formes de défaillance

suivant le degré de dégradation du système : la défaillance partielle et la défaillance complète. Leur définition est donnée ci-dessous.

Définition 3 : « Un système industriel ». C'est un processus composé par l'ensemble des transformations, et les moyens de celles-ci, opérées sur les flux entrées / sorties.

Définition 4 : « Défaillance partielle ». Une défaillance partielle correspond à une dégradation de l'aptitude d'un système à accomplir des fonctions requises. Une défaillance partielle résulte de déviations d'une ou plusieurs caractéristiques du système au-delà des limites spécifiées, telle qu'elle n'entraîne pas une disparition complète des fonctions requises. Lorsqu'une défaillance entraîne une disparition complète des fonctions du système, il s'agit d'une défaillance complète.

Définition 5 : « Défaillance complète ». Une défaillance complète est représentée par une cessation de l'aptitude d'un système à accomplir l'ensemble des fonctions requises. Un équipement défaillant nécessite une opération de maintenance pour être réparé. Les activités de dépannage n'intègrent pas d'aspect préventif. Il est bien plus intéressant de prévenir une panne avant qu'elle ne provoque une défaillance d'un équipement du système. La maintenance peut également aider à la reconstitution et à l'amélioration du système. Pour cela, elle doit prendre en considération de nombreuses contraintes comme la qualité, la sécurité, l'environnement, le coût, etc.

Une défaillance est générée par une faute. Le concept de faute est défini dans [5].

Définition 6 : « Faute ». Une faute représente une déviation non acceptable d'au moins une propriété caractéristique ou d'un paramètre du système.

Les défaillances sont généralement surveillées à l'aide de grandeurs quantitatives appelées des indicateurs. Lorsque ces indicateurs révèlent un comportement anormal, ils deviennent des symptômes. Les symptômes traduisent les effets observables des défaillances.

Définition 7 : « Un symptôme » est l'effet ou la conséquence visible d'une défaillance.

Définition 8 : « La surveillance ». C'est un dispositif passif, informationnel, qui analyse l'état du système et fournit des indicateurs. La surveillance consiste notamment à détecter et classer les défaillances en observant l'évolution du système, puis à les diagnostiquer en localisant les éléments défaillants et en identifiant les causes premières. La figure 1.1 montre les composantes de la surveillance industrielle.

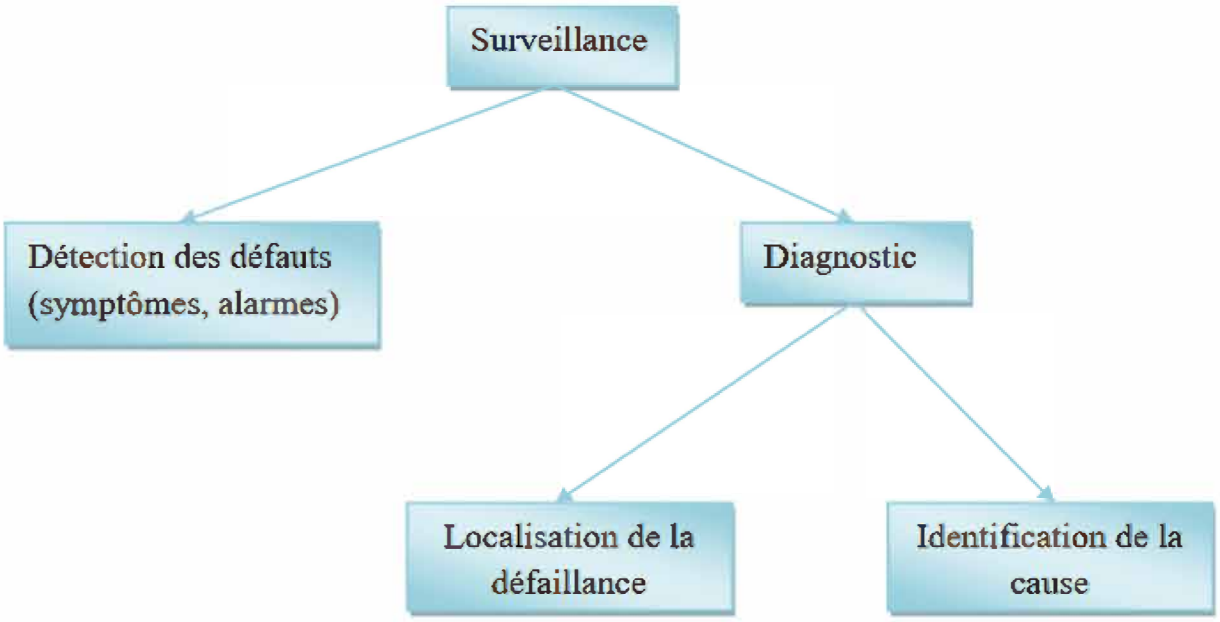

FIGURE 1.1 – Composantes de la surveillance industrielle

1.3 Objectifs de la maintenance

Dekker dans [6] décompose les objectifs de la maintenance en 4 points en fonction du domaine d'application et des priorités du décideur :

— *Assurer le fonctionnement du système :* généralement vis-à-vis d'objectifs de production. C'est la vision la plus commune de la maintenance dans le domaine du Génie Industriel. Elle se mesure au travers d'indicateurs fonctionnels tels que la fiabilité, la disponibilité, son efficacité et aussi au travers d'indicateurs qualité tels que le taux de rebut. Dans ce contexte, il est possible d'évaluer la valeur économique des résultats de maintenance au travers de ces indicateurs.

— *Maintenir la valeur économique du bien (asset management) :* grâce à la maintenance, on préserve l'état du système et on augmente sa durée d'usage. Cet objectif est particulièrement étudié dans le domaine des structures civiles pour lesquelles il peut être difficile de mesurer l'effet de maintenance et de définir la notion de défaillance. Par exemple, à quel niveau de dégradation juge-t-on une route défaillante ?

— *Assurer la sécurité des biens et des hommes :* la sécurité est le premier objectif dans les cas

où la défaillance peut causer des conséquences dramatiques, tels que le domaine nucléaire, l'aviation. L'objectif de maintenance dans ces cas est d'assurer que le niveau de risque reste inférieur à une limite stricte et appropriée. De-là, l'inspection et l'examen sont des missions essentielles qui jouent le rôle important dans le processus de maintenance.

— *Assurer le bien-être de l'humanité :* on considère le bien-être de l'Homme comme objectif dans les cas où il n'y a pas la relation directe avec les aspects économiques ou techniques, mais plutôt d'ordre psychologique, par exemple, la restauration d'œuvres d'art telles que les peintures, les statues, etc. Cet objectif est le plus difficile à quantifier.

1.4 Méthodes de maintenance

Les auteurs dans [7] et [8] ont effectué ces dernières années des recherches relatives au développement et à l'exécution de plans de maintenance dans le but de garantir la fiabilité et la sûreté de fonctionnement des systèmes. Leurs travaux définissent la maintenance comme étant une action corrective pour réparer une panne déjà existante ou une action préventive pour prévenir la défaillance et l'apparition d'une panne. Finalement, deux stratégies de maintenance ont été identifiées : la maintenance corrective et la maintenance préventive. La figure 1.2 montre ces deux types de maintenance [7].

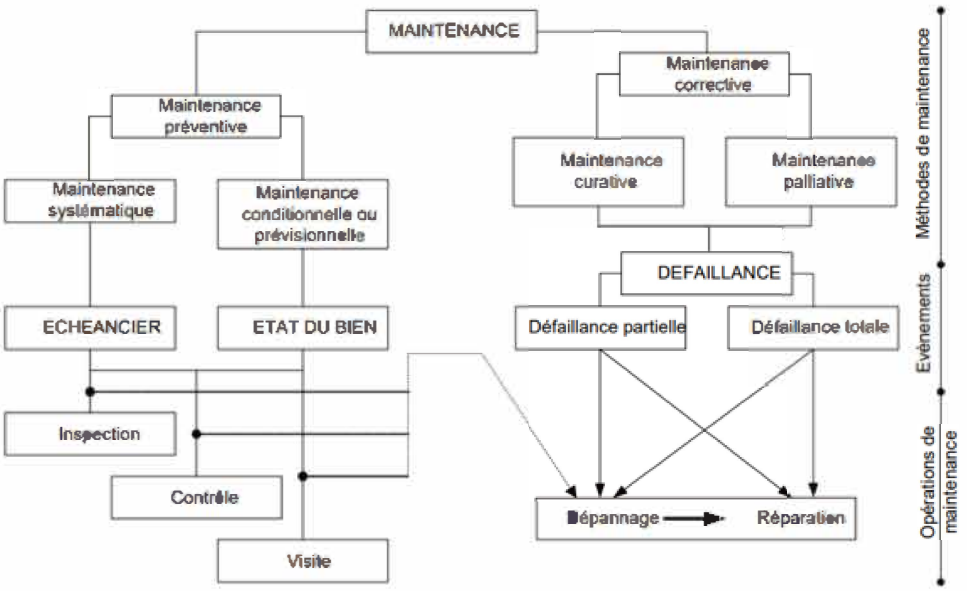

FIGURE 1.2 – Les types de maintenance

1.4.1 La maintenance corrective

La maintenance corrective est exécutée après détection d'une panne ou d'une défaillance. Ce type de maintenance est destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise. Pratiquement, c'est faire fonctionner sans entretien un système jusqu'à la « rupture ou la panne » de ce dernier. Ce type de maintenance est généralement adapté pour les équipements pour lesquels :

— les conséquences de la panne ne sont pas critiques,

— les coûts d'investissements sont faibles.

Deux formes de maintenance corrective peuvent être distinguées :

— la maintenance corrective palliative : elle est fondée sur l'action de dépannage qui permet de remettre provisoirement le matériel à un niveau de performance acceptable qui peut être inférieur au niveau optimal. L'intervention est donc à caractère provisoire, la norme AFNOR [4] la décrit comme : « Action de maintenance corrective destinée à permettre à un bien d'accomplir provisoirement tout ou une partie d'une fonction requise, appelée couramment dépannage ». La maintenance palliative est principalement constituée d'actions à caractère provisoire qui doivent être suivies d'actions curatives.

— la maintenance corrective curative : en industrie, la maintenance curative est fortement liée à la fin de vie d'une machine. En effet, on parle de maintenance industrielle curative quand un équipement tombe en panne et qu'il n'y a pas de possibilité de réparation. Les interventions dans ce type de maintenance corrective sont de caractère définitif. L'intervention qui suit la défaillance permet le rétablissement du niveau de performance optimal du matériel.

1.4.2 La maintenance préventive

Contrairement à la maintenance corrective qui attend l'occurrence de la panne pour intervenir en causant l'augmentation des coûts indirects liés à l'interruption de la production, la maintenance préventive consiste à intervenir sur un équipement avant que celui-ci ne soit défaillant. Les interventions doivent être déclenchées avant les pannes. Il est donc nécessaire d'anticiper et de réduire le risque d'occurrence des défaillances en programmant des opérations de maintenance avant leur apparition. Les stratégies développées dans cette optique intègrent le domaine de la maintenance préventive autrement dit des maintenances destinées à évaluer et/ou atténuer la dégradation et

réduire la probabilité de défaillance d'un bien. D'après la figure 1.2, la maintenance préventive se divise en deux catégories : la maintenance systématique et la maintenance conditionnelle.

1. **Maintenance systématique**

   Lorsque l'intervention de maintenance est exécutée à intervalles fixes et prédéfinis, on parle de maintenance préventive systématique. Ce type de maintenance, également connue sous le nom de maintenance planifiée, demande un suivi rigoureux du nombre d'heures ou de cycles de fonctionnement. Elle exige cependant une bonne connaissance des durées de vie et des caractéristiques de fiabilité des différents composants et sous-ensembles (ils seront remplacés systématiquement). Ce type de maintenance est souvent appliqué aux pièces critiques d'un équipement dont le coût d'indisponibilité est élevé. Cette maintenance est également appliquée si les éléments (où les pièces) à changer ne sont pas chers ou très difficilement accessibles. Cette estimation peut être réalisée par des méthodes statistiques ou au moyen de modèles de dégradation théoriques en reproduisant artificiellement les conditions d'utilisation. On pourrait prendre en exemple la réalisation d'un test de fatigue sur banc d'essais. Une fois cette durée de vie établie, on peut anticiper le remplacement de l'élément en fin de vie en prenant une marge de sécurité. Ce type d'approche permet ainsi de réduire le risque de défaillance par un entretien régulier. Toutefois, cette stratégie a un coût. Bien que le risque d'indisponibilité imprévue diminue, la maintenance peut nécessiter un arrêt programmé du bien, ce qui réduit sa disponibilité globale. De nos jours, l'activité de maintenance est soumise à des exigences de plus en plus fortes. Ces exigences ont à la fois trait à la fiabilité de systèmes toujours plus complexes et donc difficiles à maintenir, mais également à la réduction des coûts qui est devenue un enjeu majeur sur toutes les activités non porteuses de valeur ajoutée. C'est pour cette raison que le besoin d'une maintenance conditionnelle a vu le jour afin de réduire le coût de la maintenance tout en conservant ou améliorant la fiabilité des biens.

2. **La maintenance conditionnelle**

   La maintenance conditionnelle est définie comme une : « Maintenance préventive fondée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent » (selon Afnor) [4]. C'est une maintenance subordonnée à un type d'événement prédéterminé (diagnostic, capteur, signal ou mesure)

révélateur de l'état de dégradation. Avec l'arrivée des nouvelles technologies d'acquisition de données en temps réel, on observe maintenant une recrudescence des recherches portant sur les maintenances conditionnelles [9]. Ces méthodes se fondent, soit sur une expertise métier pour l'élaboration d'un modèle de dégradation, soit sur des données de fonctionnement pour l'entrainement de méthodes supervisées, semi-supervisées ou non-supervisées [10]. La maintenance préventive conditionnelle trouve sa capacité à détecter un défaut imminent et sa précision dans la prédiction de la défaillance. Elle contribue également au diagnostic des défauts car il est relativement facile d'associer un défaut spécifique dans le système aux paramètres surveillés. Comme le montre le tableau 1.1 page 17, elle peut réduire considérablement les coûts d'intervention et accroître la sûreté de fonctionnement des systèmes. Elle optimise le compromis entre les coûts de la maintenance et les performances en augmentant la disponibilité et la fiabilité tout en éliminant les activités de maintenance inutiles.

Types de maintenance	Avantages	Inconvénients
Maintenance systématique	— Réduction des pannes imprévues — Prolongation de la durée de vie — Amélioration de la fiabilité	— Coûts élevés — Interruption de la production — Réparation inutile
Maintenance corrective	— Réduction de temps d'arrêt — Prolongation de la durée de vie — Coût de maintenance maitrisé — Planification précise	— Coûts initiaux élevés — Fiabilité des données — Possibilité de faux positifs et faux négatifs
Maintenance conditionnelle	— Une réduction du nombre de pannes — Des choix ciblés pour les interventions à effectuer — Des augmentations de la performance et de la durée	— Coûts d'investissement élevés

TABLE 1.1 – Avantages et inconvénients des maintenances corrective et préventive

1.5 Optimisation de la maintenance

Le choix entre les méthodes de maintenance s'effectue dans le cadre de la politique de la maintenance et doit s'opérer en accord avec la direction de l'entreprise. Pour choisir, il faut donc être informé des objectifs de la direction, des politiques de maintenance, mais il faut connaître le fonctionnement et les caractéristiques des matériels, le comportement du matériel en exploitation, les conditions d'application de chaque méthode, les coûts de maintenance et les coûts de perte de production. Lors du choix de la méthode de maintenance, il faut arbitrer entre les performances attendues du système et les coûts que l'on est prêt à assumer. Par exemple, les économies sur les coûts de maintenance finissent généralement par coûter cher en arrêts de fonctionnement du système. Inversement, au-delà d'un certain seuil, un niveau de maintenance trop important coûte cher sans forcément apporter de supplément de performance au système. Il faut donc arbitrer entre le niveau de disponibilité des équipements que l'on souhaite garantir et le niveau de coûts directs de maintenance acceptable (personnel, matériels) comme le montre la figure 1.3 qui provient de l'ouvrage [11].

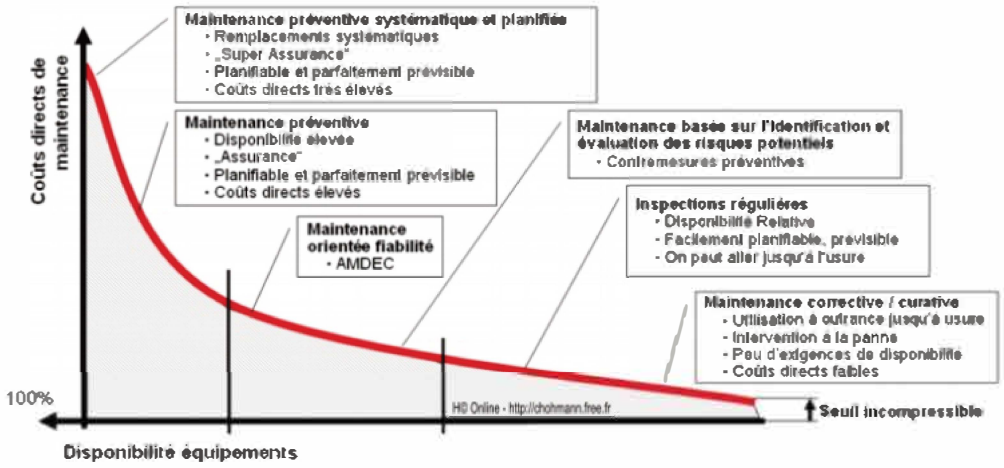

FIGURE 1.3 – Le choix de la bonne politique de maintenance

Le niveau minimum des dépenses réelles correspond à un seuil incompressible dû aux pannes imprévisibles qu'il va bien falloir réparer, quels que soient les coûts que cela engendre, sous peine d'arrêt du système.

Optimiser la maintenance consiste à garantir un certain niveau de disponibilité des équipements

du système pour un moindre coût. Afin d'éviter les coûts dus à l'arrêt système pour le réparer, il est préférable d'anticiper les pannes potentielles du système et pour les éviter.
La maintenance prévisionnelle (ou proactive) permet d'identifier les causes des futures défaillances d'un système en tenant compte de l'état réel de dégradation et d'augmenter la durée de vie du système contrairement à :

— réparer un système à chaque panne,

— considérer que les pannes sont inévitables et donc normales,

— mettre en place une maintenance systématique.

Des méthodes de diagnostic sont nécessaires de manière à détecter les signes précurseurs d'une défaillance potentielle dans le système et à identifier les causes des pannes en termes d'équipement à réparer. Des méthodes de pronostic peuvent ensuite s'appuyer sur ces informations pour évaluer et prédire les effets de ces pannes sur les autres équipements du système [12]. Les méthodes de pronostic permettent également d'améliorer la planification et l'ordonnancement des actions de maintenance.

1.6 Concepts de diagnostic et de pronostic pour la maintenance

Le diagnostic et le pronostic de pannes des systèmes d'entraînement électriques sont un enjeu majeur pour assurer la fiabilité et la sûreté de fonctionnement des outils de production notamment dans les secteurs sensibles (militaire, l'aéronautique, l'aérospatiale et nucléaire, etc.). La maintenance préventive conditionnelle se compose d'un certain nombre de modules fonctionnels : acquisition de données, traitement du signal, surveillance, diagnostic, pronostic et l'aide à la décision. En outre, une interface homme-machine peut être ajoutée afin de fournir un moyen d'afficher des informations vitales sur l'état de santé du système. Dans ce manuscrit, nous allons nous focaliser sur les deux modules diagnostic et pronostic.

1.6.1 Le diagnostic

Le diagnostic d'un système matériel désigne toute méthode permettant de déterminer si une machine est défaillante ou non et de déterminer l'origine de la panne à partir des informations relevées par observation, contrôles et tests. Le diagnostic d'un défaut, alors que le système est en cours de fonctionnement, permet d'éviter des arrêts imprévus et par la même occasion, réduit

les coûts liés à la maintenance. Par conséquent, le diagnostic des défauts est devenu un sujet de recherche qui suscite auprès des industriels un intérêt grandissant qui a débouché sur de nombreux travaux de recherche afin de développer des méthodes de diagnostic encore plus efficaces. On peut citer par exemple des méthodes analytiques, statistiques comme les applications des méthodes statistiques multivariées pour le diagnostic des défauts telles que l'analyse en composantes principales (ACP) largement rapportée dans la littérature. Un aperçu de l'utilisation de l'ACP dans le diagnostic des défauts a été donné par [13, 14]. Des méthodes fondées sur l'intelligence artificielle ont été présentées dans [15, 16]. Ou encore, des méthodes qui utilisent le principe de classification car le diagnostic d'un défaut dans un système peut être vu comme un problème de classification et d'identification [17, 18].

Méthode de diagnostic

Les méthodes de diagnostic se distinguent selon différents critères : la dynamique du procédé (discret, continu ou hybride), sa complexité, l'implémentation du diagnostic en ligne et/ou hors ligne, la nature des informations (qualitative et/ou quantitative), sa profondeur (structurelle, fontionnelle et/ou temporelle), sa distribution (centralisée, décentralisée ou distribuée), etc. Dans ce contexte, plusieurs classifications sont proposées dans la littérature [19, 20]. Ces classifications sont influencées par les terminologies et les contextes particuliers de chaque communauté et ne sont pas toujours homogènes. Nous proposons une classification des méthodes de diagnostic en deux grandes familles :

— les méthodes sans modèle mathématique qui ne nécessitent pas de connaissances accrues du système physique, mais utilisent des connaissances superficielles,

— les méthodes à base de modèles qui nécessitent une connaissance approfondie du sytème physique.

Diagnostic fondé sur un modèle physique : l'utilisation des modèles pour le diagnostic date du début des années 70. Depuis, de nombreux travaux ont été proposés [21, 22]. Ces méthodes exploitent les modèles physiques ou les modèles boîtes noires (modèles entrées/sorties).

Le diagnostic à l'aide de ces méthodes consiste à générer des indicateurs de présence de défauts dans le système que l'on appelle « résidus ». Ces modèles peuvent être classés en deux catégories

quantitatives ou qualitatives.

1. **méthodes quantitatives** :

   Ces méthodes exploitent les modèles physiques ou les modèles boîtes noires (modèles entrées/sorties). Ces méthodes nécessitent deux étapes. La première étape génère des irrégularités entre le comportement réel du système et celui estimé par le modèle. De telles irrégularités, aussi appelées résidus, reflètent les défauts potentiels du système. Chaque résidu représente une signature unique d'un défaut et donc les éléments défectueux peuvent être facilement identifiés. La deuxième étape choisit une règle de décision pour le diagnostic. Parmis les méthodes les plus utilisées, nous trouvons les méthodes à base d'observateurs [23, 24]. Il existe une grande variété de méthodes fondées sur les résidus, par exemple, les méthodes à base d'observateurs, espace de parité et les méthodes fondées sur l'estimation paramétrique.

   Observateurs : les observateurs sont généralement utilisés dans la synthèse des lois de commande des systèmes physiques. Ils sont sensés fonctionner dans le contexte d'un système sans défaut : ils sont donc choisis et calculés en fonction de critères de stabilité et de performance. Un défaut se traduit par la rupture des hypothèses d'équilibre du modèle. Il peut se révéler dans la rupture des grandeurs observées par rapport aux grandeurs mesurées. L'écart entre la sortie réelle et la sortie observée utilisé comme résidu est un indicateur de défaut et peut donc fournir des informations exploitables pour la détection et la localisation des défauts. Un résidu proche ou égal à zéro signifie que le système fonctionne normalement. Par contre, un résidu éloigné de zéro indiquera une divergence entre la sortie du système et la sortie estimée par le modèle. Cette différence permet la détection d'un défaut dans le système. La figure 1.4 donne le schéma conceptuel d'un générateur de résidus à base d'un observateur [25].

   Espace de parité : Cette technique consiste à tester la nullité d'équations de redondance entre différentes grandeurs connues caractérisant le système à surveiller. Ces équations sont généralement obtenues en éliminant, des équations modélisant le système, les grandeurs inconnues ; dans le cas où les paramètres de ce modèle sont incertains et/ou variants dans le temps, le résultat de l'élimination est critiquable. Les méthodes de type espace de parité et celles à base d'observateurs sont similaires, comme l'indique les recherches effectuées par

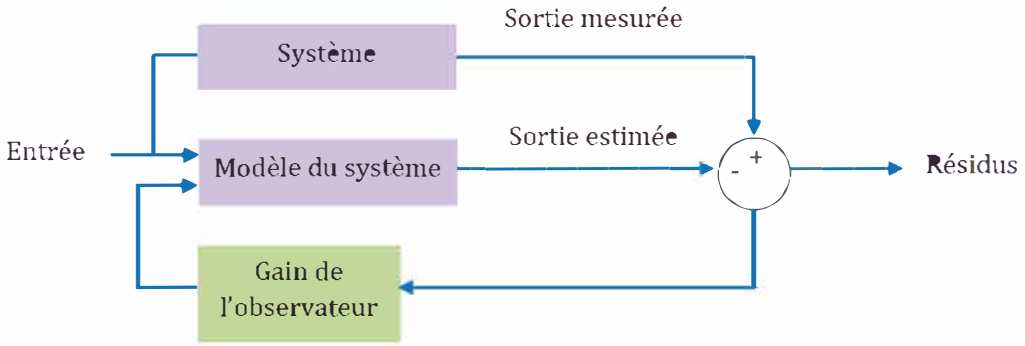

FIGURE 1.4 – Générateur de résidus à base d'un observateur

[26, 27]. La figure 1.5 donne le schéma conceptuel d'une approche de type espace de parité pour un générateur de résidus [28].

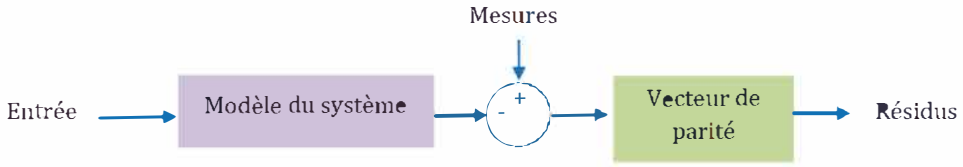

FIGURE 1.5 – Générateur de résidus dans l'espace de parité

Estimation paramétrique : alors que les deux premières catégories sont bien adaptées au diagnostic de défauts de capteurs et d'actionneurs, la troisième catégorie est a priori mieux adaptée au diagnostic des défauts internes du procédé. L'idée directrice de la surveillance par estimation paramétrique est qu'un défaut interne doit se traduire par une modification des paramètres du système. De nombreuses techniques ont été développées à partir de modèles de type boîte noire à représentation discrète [29] en association avec le concept de rupture de modèle pour détecter l'apparition d'un défaut. Une autre approche consiste à utiliser des modèles de type boîte grise à représentation continue [30] pour détecter et localiser un défaut à partir de la surveillance de paramètres physiques. Toutefois, un des inconvénients majeurs de cette méthode réside dans la nécessité d'avoir un système physique excité en permanence. La figure 1.6 illustre la description schématique d'une estimation paramétrique.

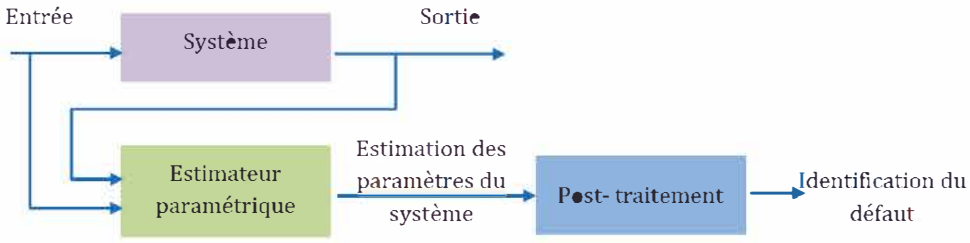

FIGURE 1.6 – Description schématique d'une estimation paramétrique

2. **Méthodes qualitatives :** les chercheurs ont proposé un raisonnement qualitatif (ou semiqualitatif) fondé sur l'établissement de relations causes à effets. Le diagnostic est un système causal vu qu'il consiste à établir des hypothèses sur les composants défaillants qui sont l'origine du dysfonctionnement observé. Le raisonnement qualitatif met en évidence le lien entre un composant et les formules décrivant son comportement. Nous pouvons citer les méthodes de graphes causaux qui consiste à déterminer les composants défectueux qui peuvent expliquer le fonctionnement anormal observé. Le diagnostic fondé sur un graphe causal consiste à rechercher la variable source dont la déviation est suffisante pour expliquer toutes les déviations détectées sur d'autres variables. La deuxième méthode est la logique floue. Cette méthode est une théorie mathématique introduite par [31]. Elle permet de tenir compte des incertitudes et permet une fusion des informations. L'idée de l'approche floue est de construire un dispositif, appelé système d'inférences floues, capable d'imiter les prises de décision d'un opérateur humain à partir des règles verbales traduisant ses connaissances relatives à un processus donné. Trouver un lien mathématique entre un défaut et ses symptômes est souvent difficile. Cependant, en s'appuyant sur leur expérience, les opérateurs humains sont capables de déterminer l'élément défaillant qui est à l'origine des symptômes observés. Ce type de savoir peut être exprimé à l'aide de règles d'une condition : SI condition ALORS conclusion. Où la partie condition comporte les symptômes observés et la partie conclusion l'élément défaillant.

Méthodes sans modèle : pour certaines applications industrielles, la conception d'un modèle mathématique est difficile, voire impossible à obtenir, à cause des nombreuses reconfigurations intervenant dans le processus de production ou de la complexité des phénomènes mis en jeu. Dans ce cas, on a recours à des méthodes qui ne nécessitent aucune connaissance approfondie du

procédé. Deux classes, dans ce type d'approches, peuvent se présenter :

— les méthodes quantitatives ou appelées méthodes à base de connaissances, ‘

— les méthodes qualitatives ou méthodes fondées sur le traitement de données,

1. **Méthodes à base de connaissances :** ces méthodes exploitent la connaissance des experts sur le système, une connaissance acquise durant la phase de conception et/ou d'un historique des dysfonctionnements du système.

   Ces méthodes sont appellées « systèmes experts ». Elles sont mises en œuvre lorsque la majorité des mesures sont indisponibles et lorsque la construction du modèle s'avère difficile. Le système expert est un système informatique destiné à résoudre un problème précis à partir d'une analyse, d'une représentation des connaissances et du raisonnement d'un ou de plusieurs spécialistes de ce problème. Ils utilisent une information heuristique pour lier les symptômes aux défauts. A partir de l'ensemble des symptômes à sa disposition, il déduit toutes les conclusions possibles, élabore de nouvelles hypothèses et approfondit son diagnostic en exploitant des informations supplémentaires collectées sur le système à diagnostiquer. Les systèmes experts sont composés de deux parties indépendantes : une base de connaissances et un moteur d'inférence. La base de connaissances peut être subdivisée en deux groupes : la base de faits et la base de règles. La base de faits représente l'état du système observé. La base de règles est un ensemble de règles logiques qui permettent de déduire, d'inférer de nouveaux faits à partir d'autres faits déjà établis dénommés les prémisses. Le moteur d'inférence a la lourde tâche de déduire tous les faits susceptibles d'être déduits à partir de la base de connaissances.

2. **Méthodes fondées sur le traitement de données :** ces méthodes exploitent la connaissance des experts sur le système, une connaissance acquise durant la phase de conception et/ou d'un historique des dysfonctionnements du système. Ce sont des méthodes d'analyse de données et d'aide à la décision pour les experts. Ces techniques dites « systèmes experts » font appel à des méthodes statistiques comme l'analyse en composantes principales (ACP), la classification et des méthodes fondées sur l'intelligence artificielle comme les réseaux de neurones.

   Analyse en composantes principales (ACP) : ces méthodes ont été largement rapportées dans la littérature. Un aperçu de l'utilisation de l'ACP dans le diagnostic des défauts a été

donné par plusieurs chercheurs comme Kresta et al [13]. L'analyse en composantes principales est une méthode de compression de données. Elle réduit un ensemble de données à quelques éléments d'information, mais qui expliquent l'essentiel des variations du système. L'ACP combine des outils de l'algèbre linéaire (décomposition en valeurs propres) et l'analyse statistique (variance-covariance) pour construire un modèle représentatif qui peut être utilisé pour le diagnostic des défauts.

Réseau de neurones : le réseau de neurones peut être assimilé à un système informatique constitué d'un nombre de processeurs élémentaires interconnectés entre eux qui traite l'information qui lui arrive à partir des signaux extérieurs. Il est capable d'approcher une fonction non-linéaire dont on ne connait que quelques points. Les réseaux de neurones, utilisés pour le diagnostic des défauts, peuvent être classés selon : l'architecture du réseaux, et la méthode d'apprentissage. Il y a eu beacoup de travaux qui abordent le problème du diagnostic des défauts par des réseaux de neurones [32]. Leur force réside dans l'aptitude à apprendre et à capturer les relations entre l'entrée du réseau de neurones et sa sortie même si ces relations sont inconnues ou difficiles à décrire. Cependant, leur principal inconvénient réside dans l'acquisition et la disponibilité des données d'apprentissage.

Reconnaissance de forme ou classification : l'objectif d'une méthode de reconnaissance de forme est la classification automatique d'objet suivant sa ressemblance par rapport à un objet de référence. Le diagnostic d'un défaut dans un système peut être interprété comme un problème de classification [33]. Une classe est formée par l'ensemble d'observations caractérisant une situation ou un mode de fonctionnement de processus : par exemple, la classe C1 peut être liée au fonctionnement normal du procédé, la classe C2 pour le fonctionnement dégradé et la classe C3 pour le fonctionnement défaillant. Le diagnostic consiste à associer toute nouvelle observation à une classe.

1.7 Analyses comparatives des méthodes de diagnostic

Comparer les méthodes de diagnostic est une tâche très difficile. En effet, la décision de choix d'une méthode de détection de défaut dépend de plusieurs facteurs : connaissance à priori disponible sur le système, présence ou non d'un modèle mathématique de processus à diagnostiquer,

type de défauts à détecter, présence ou non d'entrées inconnues (bruit, incertitude), non linéarité, système en boucle fermée ou ouverte etc. Par exemple, pour les systèmes électriques et mécaniques, il est simple de concevoir un modèle mathématique. Le choix des méthodes de diagnostic à base de modèles sera préféré. Par contre, pour les processus industriels chimiques, la modélisation est difficile à mener car même si un modèle mathématique est obtenu, il est complexe. Dans ce cas, les méthodes de détection de défauts sans modèle de référence seront mieux adaptées. Isermann dans [20] a proposé une étude comparative entre les méthodes de diagnostic par redondance analytique. Il est à noter que de nombreuses méthodes sont apparues en combinant les avantages de chaque stratégie de diagnostic. Quelques approches combinées sont présentées dans [20]. Le tableau 1.2 propose une analyse comparative synthétique des approches de détection et de diagnostic que nous venons de citer. Il ressort de cette comparaison que les méthodes à base d'un modèle nécessitent un modèle analytique pour représenter un système. Dans le cas réel, la disponibilité d'un modèle analytique est souvent irréaliste ou inexacte en raison de la complexité du système, ceci peut être causé par des modèles ne reflétant pas de façon précise le comportement du système. Les méthodes fondées sur le traitement de données, quant à elles, ne nécessitent pas un modèle analytique pour diagnostiquer un système. Seule la disponibilité des données de mesure est nécessaire. Par conséquent, ces méthodes sont une alternative dans le cas de la non disponibilité d'un modèle analytique. Dans la suite de cette thèse, nous nous intéresserons essentiellement aux méthodes fondées sur le traitement de données, plus précisément, au diagnostic par classification supervisée.

Critère / Méthode	Diagnostic rapide	Identification	Robustesse	Adaptation	Facilité d'exploitation	temps de calcul
Observateurs	OUI	OUI	OUI	NON	NON	NON
Espace de parité	OUI	OUI	OUI	NON	NON	NON
Estimation paramétrique	NON	OUI	Dépend de la situation	NON	NON	NON
Systèmes experts	OUI	OUI	OUI	NON	OUI	OUI
ACP	OUI	OUI	OUI	NON	NON	OUI
Réseau de neurones	OUI	OUI	Dépend de la situation	OUI	NON	Dépend de la situation
Classification	OUI	OUI	Dépend de la situation	OUI	OUI	Dépend de la situation

TABLE 1.2 – Comparaison des différentes méthodes utilisées pour le diagnostic des défauts

1.7.1 Le pronostic

La maintenance préventive conditionnelle nécessite une approche prédictive. Le processus de pronostic intervient dans l'intervalle entre l'apparition du défaut et la défaillance. Il consiste à prédire la durée de vie résiduelle du système en considérant les informations passées (historiques) et/ou présentes (état courant), voire futures (missions futures du système). L'organisation internationale de normalisation (ISO) 13381-1 [ISO13381] a défini le pronostic comme étant une estimation de la durée de vie restante avant l'apparition d'un défaut et la détection de l'imminence d'un ou plusieurs défauts. Le pronostic détermine si un défaut est imminent et estime le temps restant avant son apparition. Les progrès d'aujourd'hui dans la maintenance préventive conditionnelle ont contribué à améliorer le pronostic sur l'état futur des systèmes. Cette avancée permet non seulement de réduire les coûts liés à la maintenance mais aussi d'augmenter l'efficacité des interventions. La figure 1.7 montre la différence entre le diagnostic et le pronostic.

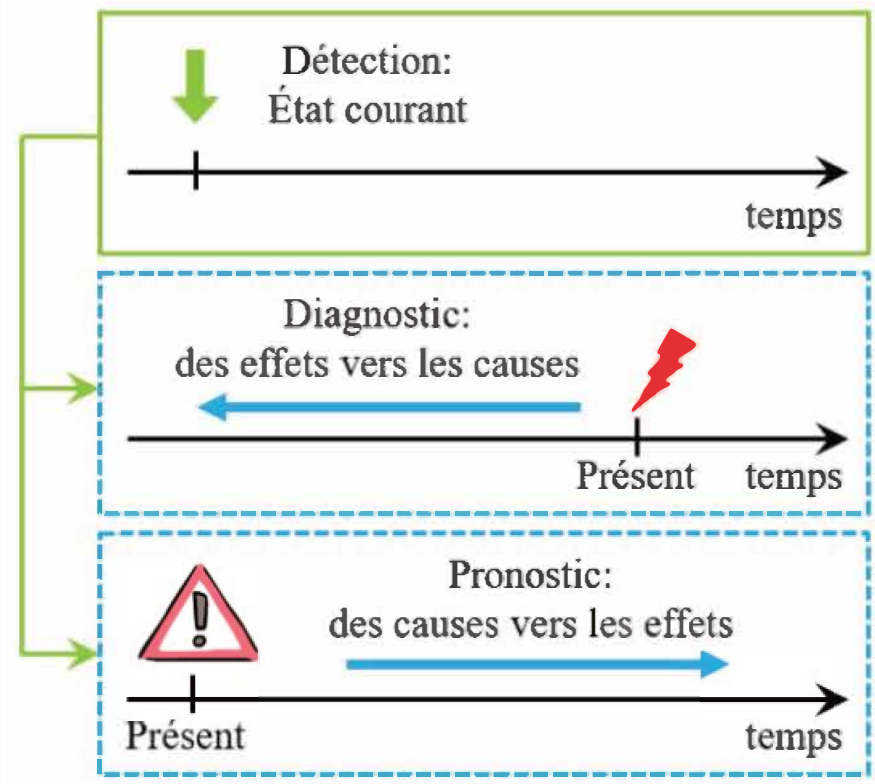

FIGURE 1.7 – La différence entre le diagnostic et le pronostic

Méthodes de pronostic

Dans la littérature, de nombreuses classifications des méthodes et outils de pronostic de défaillance ont été proposées. Elles se différencient en matière du résultat attendu et de la nature des informations utilisées. Sikorska et al [34] presentent un résumé des différentes classifications des méthodes. La classification des méthodes pour le pronostic selon la connaissance que l'on a sur le système est identique à celle des méthodes pour le diagnostic. Beacoup de chercheurs ont classé les méthodes sous forme pyramidale selon la complexité de main d'œuvre, le coût, la précision des résultats obtenus et l'applicabilité des approches. La figure 1.8 illustre les trois classes de méthodes de pronostic qui reposent sur différents niveaux de connaissance disponible.

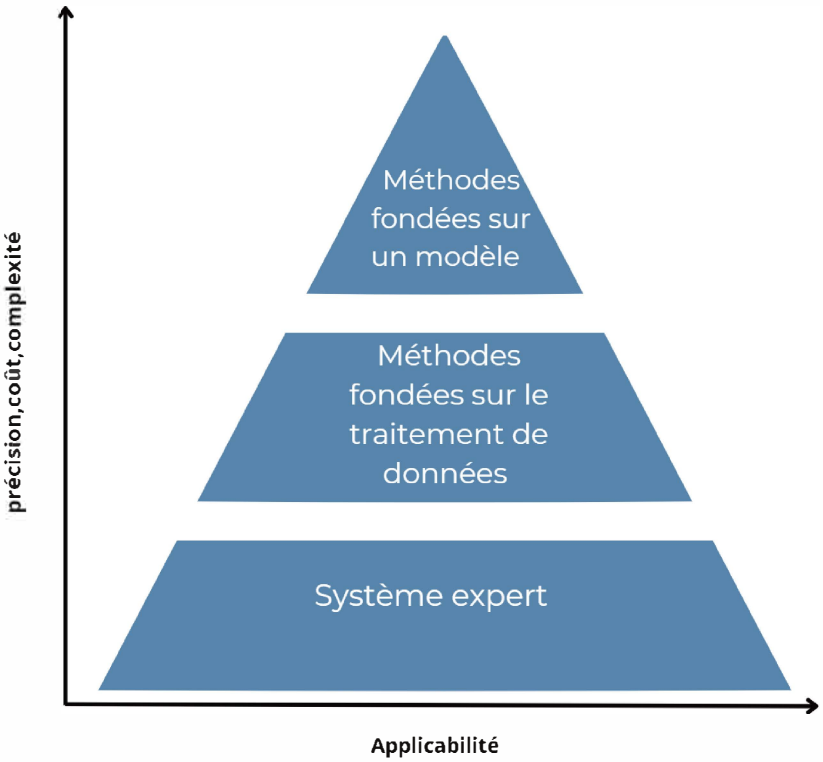

FIGURE 1.8 – Différentes méthodes de pronostic

Approches fondées sur les modèles physiques : ce type approche est fondé sur la compréhension des principes et des mécanismes physiques de dégradation se produisant sur un système. Cette connaissance permet de construire un modèle de comportement d'un système intégrant l'évolution dynamique de la dégradation. Dans ce cas, la dégradation est une variable continue dont l'évolution est décrite par un processus déterministe. Le premier avantage d'un pronostic

fondé sur un modèle physique est qu'il ne nécessite pas de données historiques pour estimer l'état futur du système une fois que le modèle est connu. Un autre de ces avantages est qu'il permet de prendre en compte la dégradation causée par des conditions environnementales telles que les charges thermiques, l'humidité, les vibrations et les chocs. Un de ces inconvénients est qu'une représentation mathématique d'un système nécessite une connaissance approfondie des processus physiques sous-jacents qui conduisent à la défaillance du système [35]. Un autre de ces inconvénients est qu'il est difficile de créer des modèles représentant les multiples processus physiques qui se produisent dans le système [36].

Approches fondées sur les données

Le pronostic guidé par les données s'appuie sur l'exploitation de symptômes ou bien d'indicateurs de la dégradation dont l'évolution future est déterminée grâce à une méthode statistique. Dans cette approche, la surveillance du système est effectuée à l'aide de paramètres de surveillance qui indiquent la dégradation du système. Les données recueillies sont d'abord analysées pour détecter les anomalies. Ensuite, l'évolution de la tendance des paramètres de surveillance est utilisée pour estimer le temps restant avant la défaillance du système en utilisant des algorithmes de prédiction.

Approches fondées sur l'expérience :

Une approche de pronostic fondée sur l'expérience repose sur un modèle de pronostic obtenu à partir d'une connaissance des données historiques des défaillances ou les recommandations de conception des composants dans des conditions opérationnelles similaires afin de déterminer la probabilité de défaillance à un moment donné dans le futur. Le principal inconvénient de cette approche est qu'il est difficile de disposer d'un historique d'expériences passées représentatif de toutes les conditions d'utilisation des systèmes. Les résultats de pronostic sont moins précis que ceux fondés sur un modèle et sur les données, en particulier quand les conditions de fonctionnement sont variables.

1.8 Conclusion

Ce chapitre a permis de présenter le concept de maintenance et son importance pour le bon fonctionnement de tout système. Il présente dans un premier temps les principales stratégies de

maintenance appliquées dans l'industrie ainsi que leurs avantages et inconvénients. En particulier, la maintenance corrective peut engendrer une indisponibilité importante du système ainsi que des problèmes de sécurité. La maintenance préventive systématique peut s'avérer très coûteuse si elle ne tient pas compte de l'état de dégradation des composants du système, alors que la maintenance préventive conditionnelle permet de contrôler l'état de dégradation d'une pièce avant de la remplacer. Les composants ne sont donc plus changés systématiquement indépendamment de leur état de santé. Dans une seconde partie, nous avons passé en revue les méthodes actuellement utilisées dans le diagnostic et le pronostic des défauts dans les systèmes industriels. Le diagnostic implique la détection et l'analyse de la cause ou la nature du défaut, alors que le pronostic calcule ou prédit l'état futur à la suite d'une analyse sur des données pertinentes et disponibles. Une méthode de diagnostic et de pronostic efficace peut conduire à la réduction des arrêts imprévus et à l'augmentation de la fiabilité et la sûreté de fonctionnement des systèmes et par conséquent réduit les coûts de la maintenance. L'état de l'art proposé dans le premier chapitre permet de constater que plusieurs méthodes existent pour supporter l'activité de diagnostic et du pronostic : méthodes fondées sur un modèle, méthodes fondées sur l'expérience et méthodes orientées données. Les comparaisons effectuées entre ces méthodes nous amènent à privilégier les méthodes orientées données. A ce stade, notre travail s'intégrera dans les stratégies de maintenance conditionnelle en se fondant sur les méthodes orientées données et plus précisément sur les méthodes de classification supervisée en utilisant les Modèles de Markov Cachés. Dans le chapitre suivant, nous allons faire un rappel sur les MMC. Nous allons explorer l'intersection des MMC et des applications de maintenance conditionnelle. Nous présentons une bibliographie des travaux existant dans les contextes du diagnostic et du pronostic des défauts fondés sur les MMC.

Chapitre 2

Les Modèles de Markov Cachés

2.1 Introduction

Dans le chapitre précédent, nous avons cité les différentes méthodes utilisées dans le diagnostic et le pronostic des pannes industrielles. Nous sommes arrivés à la conclusion que dans notre travail, nous utilisons les méthodes orientées données. Ces dernières représentent l'une des meilleures alternatives dans le cas de la non disponibilité d'un modèle physique afin de diagnostiquer et pronostiquer l'état de santé du système. Dans ce chapitre, nous nous intéressons à ces méthodes et plus précisément à la classification et l'identification d'une séquence d'observations dans une classe représentant un mode de fonctionnement du système en utilisant les Modèles de Markov Cachés (MMC). Ce chapitre sera constitué comme suit : dans un premier temps, nous allons présenter les MMC, ainsi que leurs paramètres. Ensuite, nous allons aborder les méthodes d'apprentissage des MMC, nous allons explorer l'intersection des MMC et des applications de maintenance conditionnelle. Nous présentons une bibliographie des travaux existant dans les contextes du diagnostic et du pronostic de défauts. Enfin, nous allons nous focaliser sur la problématique des multi-classes qui est le cœur de cette thèse.

2.2 Les Modèles de Markov Cachés (MMC)

Les Modèles de Markov Cachés (MMC) ont une longue histoire derrière eux. Les premiers travaux sont apparus en 1913 par A. Markov. Shannon conçoit la théorie de l'information en utilisant les chaines de Markov en 1948 [37], mais ce n'est qu'à partir de 1966 avec les travaux de Baum et al

[38] que les algorithmes basiques permettent l'estimation des paramètres des modèles. En 1970, l'appellation « Modèles de Markov Cachés » ou « Chaines de Markov Cachées » ou en anglais « Hidden Markov Models » sont apparus par L.P Neuwirth [39]. A partir de 1975, les MMC sont utilisés dans de nombreux domaines, tels que la reconnaissance automatique de la parole [40], la reconnaissance de l'image [41], de la météorologie ou le climat [42]. Les MMC ont été utilisés dans la biologie [43], jusqu'en, médecine comme pour le diagnostic du cancer [44].

Les MMC ont été utilisés dans la maintenance et surtout appliqués dans la maintenance conditionnelle. Pour cela, nous avons effectué des recherches bibliographiques avec les mots clés MMC et maintenance sur les bases de données (SCOPUS et ELSEVIER). Une visualisation du réseau a été effectuée grâce au logiciel VOSviewere. Elle contient deux éléments : les objets et leurs relations. Les objets peuvent représenter des auteurs, des publications, des revues, des pays ou encore des mots-clés représentés par des nœuds. Dans notre cas, nous avons sélectionné les mots clés. Les relations quant à elles sont représentées par des lignes entre les nœuds. La visualisation du réseau des résultats de notre recherche bibliométrique est représentée sur la figure 2.1.

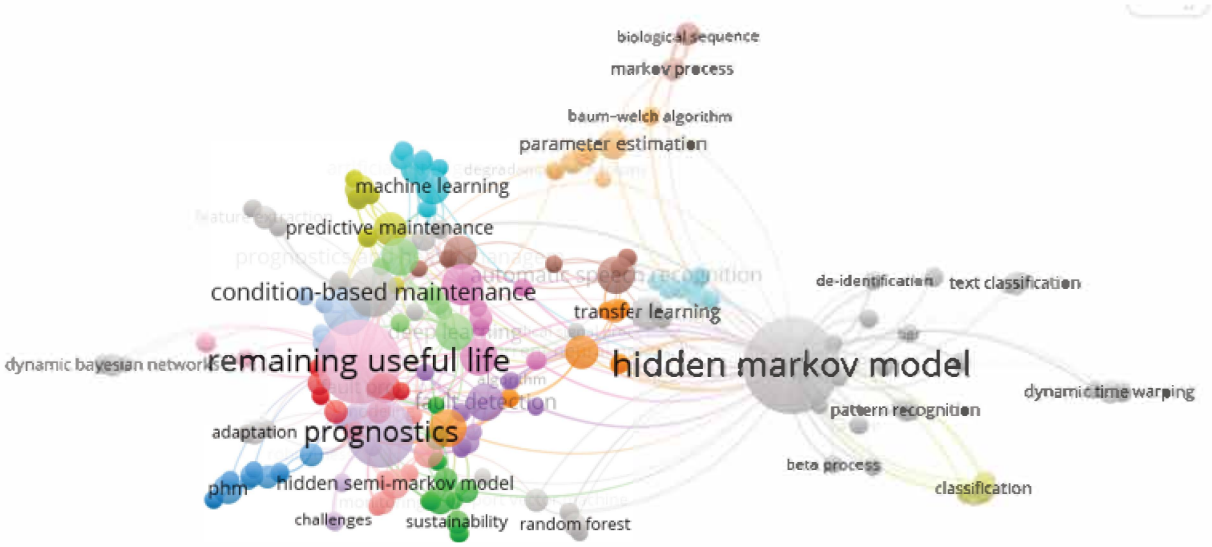

FIGURE 2.1 – Visualisation réseau

D'après cette figure, nous constatons que les grappes sont bien définies et distinctes dans notre visualisation. Globalement, nous remarquons beaucoup de relations entre les Modèles de Markov Cachés (MMC), le diagnostic et le pronostic. En effet, de nombreux travaux ont été réalisés dans

le domaine du pronostic des pannes. Pour cela nous avons effectué une étude pour analyser la distribution des travaux sur la maintenance au fil des années (voir figure 2.2).

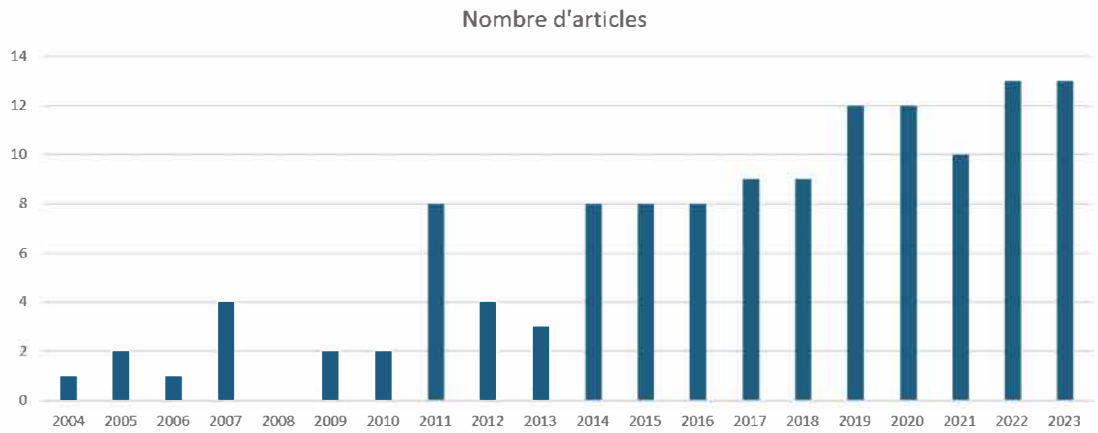

FIGURE 2.2 – Evolution des travaux qui utilisent les MMC pour le pronostic et le diagnostic

Ce graphique montre que l'étude de la maintenance en utilisant les MMC est récente et ne cesse d'évoluer. En effet de nombreux chercheurs ont utilisé les MMC afin de réaliser des diagnostics [45] ainsi que pour prédire les pannes industrielles et estimer la durée de vie des systèmes industriels [46]. Certains chercheurs ont combiné d'autres méthodes avec les MMC afin d'améliorer les résultats [47]. Dans notre travail, nous nous focalisons sur l'apprentissage supervisé des MMC afin d'améliorer le diagnostic des systèmes industriels.

2.2.1 Définition des MMC

Les processus stochastiques, notamment markoviens dans notre cas, sont des outils importants en théorie de probabilités. L'utilisation de tels processus suggère que les états sont les seules observations de la chaîne. Il serait judicieux de se demander quels types de modèles peut-on utiliser dans des situations où ces états ne sont pas directement observables, mais produisent des observations particulières «symboles». De ce fait, on s'intéresse aux MMC et aux problèmes qu'ils permettent de résoudre. Dans les processus markoviens, les observations sont les états du processus. Pour les MMC, c'est bien plus complexe. En effet, dans un MMC discret, on ne peut observer directement les états du processus mais des symboles générés par les états selon certaines lois de probabilité. Ainsi, à partir d'une séquence de symboles, il n'est pas évident de connaître la trajectoire (séquence d'états) par laquelle est passée le processus. D'où le nom de Modèles

de Markov Cachés (MMC) ou en anglais "Hidden Markov Models" (HMM). Selon Rabiner [40] et Fox [48], c'est un automate à états cachés qui est constitué d'une variable non observable. Celle-ci représente l'état du système à modéliser. Seule la variable de sortie est observable. Cela nous permet d'avoir une séquence d'observations en sortie de l'automate ; à partir de maintenant, nous parlerons simplement de symboles représentant ces observations.

2.2.2 Présentation des paramètres des MMC

Dans cette partie, nous allons décrire les éléments et les mécanismes des MMC que nous allons utiliser dans cette thèse. Dans le cas d'un processus discret, un Modèle de Markov Caché noté $\lambda = (A, B, \pi)$ est défini par les paramètres suivants :

— N le nombre d'états dans le modèle,

— M le nombre de symboles d'observation,

— $S = \{S_1, S_2, ..., S_N\}$: l'ensemble des différents états,

— $V = \{V_1, V_2, ..., V_M\}$: l'ensemble discret de symboles d'observation possibles.

— $Q = \{q_1, q_2, ..., q_T\}$: la séquence d'états avec T la longueur de la séquence, l'état à l'instant t est q_t,

— T la longueur de la séquence d'observations,

— $\{O\}$=$\{O_1, O_2, ..., O_T\}$: la séquence d'observations où chaque observation O_t est l'un des symboles de V,

— A = $\{a_{ij}\}$: la distribution de probabilité de transition d'état tel que :

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N \tag{2.1}$$

Il s'agit d'une matrice N x N dont la somme des lignes est égale à 1,

$$A = \begin{pmatrix} a_{11} & ... & a_{1N} \\ a_{21} & ... & a_{2N} \\ a_{N1} & ... & a_{NN} \end{pmatrix} \tag{2.2}$$

— B=$\{b_j(k)\}$: la distribution de probabilité des symboles d'observation dans l'état j tel que :

$$b_j(k) = P[V_k a_t \ t | q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M \tag{2.3}$$

Il s'agit d'une matrice de taille NxM,

$$B = \begin{pmatrix} b_1(1) & b_1(2) & ... & b_1(M) \\ b_2(1) & b_2(2) & ... & b_2(M) \\ b_N(1) & b_N(2) & ... & b_N(M) \end{pmatrix} \tag{2.4}$$

— π : distribution d'état initial tel que

$$\pi = \{\pi_i\} \tag{2.5}$$

Il s'agit du vecteur de probabilités inital de taille 1xN

$$\pi = [\pi_1, \pi_2, ..., \pi_N] \tag{2.6}$$

Les relations de dépendance entre les différentes variables aléatoires d'un MMC sont représentées sur la figure 2.3. La spécification d'un MMC implique le choix du nombre d'états N et du nombre de symboles discrets M, ainsi que la spécification de trois densités de probabilité A,B et π.

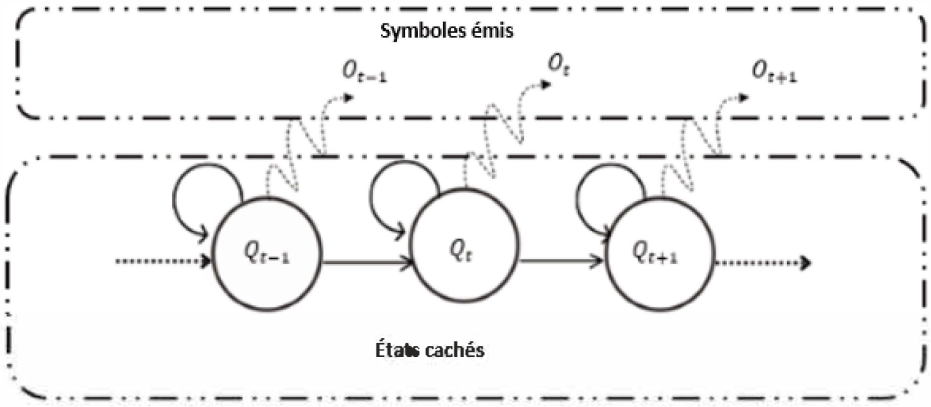

FIGURE 2.3 – Relations de dépendance entre les variables aléatoires d'un MMC

2.2.3 Principales architectures des MMC

En fonction de la spécificité du système à modéliser, la topologie du MMC peut être adaptée. Plusieurs types de topologie peuvent être utilisés. Par exemple, un modèle ergodique (figure 2.4 (a)) ou un modèle gauche-droite (figure 2.4 (b)). Le modèle ergodique n'est soumis à aucune restriction, toutes les transitions possibles entre les états de la chaîne sont autorisées. Cela est rendu possible en ne contraignant aucune des valeurs a_{ij} à être nulle. Les modèles gauche-droite permettent de mieux s'adapter aux événements qui sont orientés. Ces types de topologie sont souvent utilisés dans le domaine de la reconnaissance de caractères ou de mots. Définir une topologie signifie, seulement, mettre certaines probabilités de transition à zéro.

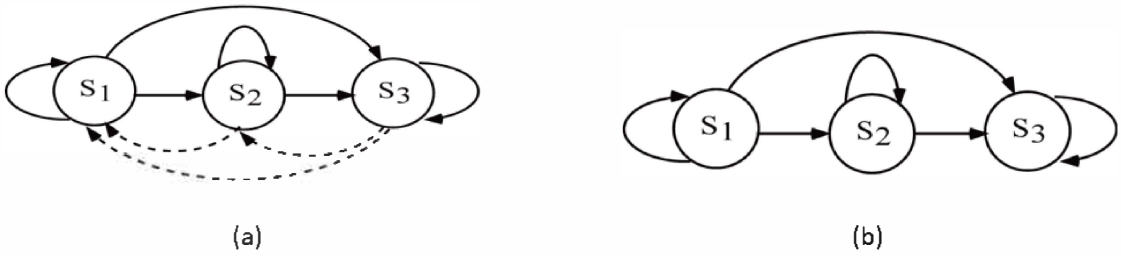

FIGURE 2.4 – Différentes topologies de MMC ((a) ergodique, (b) gauche-droite)

2.2.4 Trois problèmes fondamentaux pour les MMC

Les MMC trouvent leur intérêt dans la résolution des trois problèmes fondamentaux que sont l'évaluation, le décodage et l'apprentissage [40].

— Problème 1 : l'évaluation :

Compte tenu de la séquence d'observations $\{O\}=\{O_1,O_2,\ldots,O_T\}$ et le modèle $\lambda=$ (A, B, π), comment calculer P(O|λ), la probabilité de la séquence d'observations ? Étant donné un modèle et une séquence d'observations, comment calculer la probabilité que la séquence observée ait été produite par le modèle ? Nous pouvons également considérer que le problème consiste à évaluer dans quelle mesure un modèle donné correspond à une séquence d'observations donnée. Ce dernier point de vue est extrêmement utile. Par exemple, si nous considérons le cas où nous essayons de choisir entre plusieurs modèles concurrents, la solution du problème 1 permet de choisir le modèle qui correspond le mieux aux observations. Les algorithmes Forward et Backward sont utilisés pour résoudre ce genre de problème.

— Problème 2 : compte tenu de la séquence d'observations $\{O\} = \{O_1, O_2, \ldots, O_T\}$, comment choisir une séquence d'états qui soit optimale dans un sens donné ?

Ce problème est celui dans lequel nous essayons de découvrir la partie cachée du modèle, c'est-à-dire de trouver l'état "correct". Il devrait être clair que, sauf dans le cas des modèles dégénérés, il n'y a pas de séquence d'états "correcte" à trouver. C'est pourquoi, dans la pratique, nous utilisons généralement un critère d'optimalité pour résoudre ce problème. L'algorithme de Viterbi est généralement utilisé pour résoudre ce problème.

— Problème 3 : comment ajuster les paramètres du modèle λ = (A, B, π) pour maximiser $P(\{O\}|\lambda)$?

Ce problème consiste à essayer d'optimiser les paramètres du modèle afin de décrire au mieux la façon dont une séquence d'observations donnée se produit. Les séquences d'observations utilisées pour ajuster les paramètres du modèle forment le corpus d'apprentissage. Le problème de l'entraînement ou apprentissage est crucial pour la plupart des applications des MMC, puisqu'il nous permet d'adapter de manière optimale les paramètres du modèle aux données d'apprentissage observées, c'est-à-dire aux données d'apprentissage de l'utilisateur. L'un de ces algorithmes est appelé Baum-Welch. Il considère tous les chemins pour aller/venir vers/depuis les états. Une autre solution pour entraîner les MMC consiste à utiliser l'algorithme de Viterbi pour réestimer les paramètres du modèle. Dans ce cas, seul le meilleur chemin est considéré pour la réestimation.

Dans la section suivante, nous présentons les algorithmes utilisés pour résoudre les trois problèmes fondamentaux des MMC.

2.2.5 Les algorithmes de base

Algorithme Forward

Nous souhaitons calculer la probabilité de la séquence d'observations $\{O\} = \{O_1, O_2, \ldots, O_T\}$ étant donné le modèle λ, c'est-à-dire P($\{O\}|\lambda$). La façon la plus simple de le faire est d'énumérer toutes les séquences d'états possibles de longueur T (le nombre d'observations). Considérons une telle séquence d'états fixes.

$$Q = \{q_1, q_2, ..., q_T\} \tag{2.7}$$

Où q_1 est l'état initial. La probabilité de la séquence d'observations $\{O\}$ pour la séquence d'états de 2.7 est la suivante :

$$P(\{O\}|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) \tag{2.8}$$

Dans les MMC, l'algorithme forward délivre deux informations $P(\{O\}|\lambda)$ et $\alpha_t(i)$

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda) \tag{2.9}$$

Cette variable exprime la probabilité d'avoir généré la séquence $\{O\} = \{O_1, O_2, \ldots, O_t\}$ en partant de l'état initial et en arrivant à l'état S_j à l'instant t. Cette variable peut se calculer de manière récursive :

1. Initialisation :

$$\alpha_1(i) = \pi_i.b_i(O_1), 1 \leq i \leq N \tag{2.10}$$

2. Récursion :

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N}(\alpha_t(i).a_{ij})].b_j(O_{t+1}), 1 \leq t \leq T-1 \ et \ 1 \leq j \leq N \tag{2.11}$$

Finalement

$$P(\{O\}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.12}$$

On peut alors proposer une version complète de l'algorithme Forward (voir algorithme 1).

Algorithme 1 : Forward

Données : $\{O\}= \{O_1, O_2, ..., O_T\}$, λ=(A,B,π)

pour *i=1 à N* **faire**
  $\alpha_1(i) = \pi_i b_i(O_1)$;
fin
pour *j=1 à N* **faire**
  α_{t+1}(j)=$[\sum_{i=1}^{N}(\alpha_t(i)a_{ij}]b_j(O_{t+1})$
fin
$P(\{O\}|\lambda) = \sum_{i=1}^{N} \alpha_T$

On place dans la première 'ligne' α_1, la probabilité d'obtenir l'état caché i sachant que l'on a observé le symbole O_1 (comme le montre l'équation 2.10, avec π_i la probabilité d'avoir l'état i en premier, et $b_1(O_1)$ la probabilité d'observer O_1 lorsque l'état i est 'apparu'). Par la suite,

l'algorithme détermine la t ième ligne suivante, en s'appuyant sur la t-1 ième ligne de la matrice α. L'induction permet ainsi d'obtenir $\alpha_{t+1}(j)$, tel que $\alpha_{t+1}(j)$ soit égale à la somme des probabilités d'avoir observé les t premiers symboles suivant tous les chemins des états cachés possibles, et de passer à l'état j, en observant le symbole O_{t+1} à l'instant t+1. La matrice α ainsi obtenue condense les calculs.

Algorithme Backward

Cet algorithme délivre une seule information $\beta_t(i)$, cette variable peut se voir comme la probabilité d'observer la suite partielle $\{O\} = \{O_{t+1}, O_{t+2}, \ldots, O_T\}$ qui commence à l'instant t à l'état S_i et se termine à l'instant T :

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T, q_t = S_i|\lambda) \tag{2.13}$$

Cet algorithme permet aussi de calculer $P(O|\lambda)$ puisque

$$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i) \tag{2.14}$$

L'algorithme Backward est conçu de manière similaire à l'algorithme Forward. Une seule chose diffère : la séquence est scrutée de la fin O_T vers le début O_1. La version complète de Backward est proposée dans l'algorithme 2.

Algorithme 2 : Backward

Données : $\{O\} = \{O_1, O_2, ..., O_T\}$, λ=(A,B,π)
pour *i=1 à N* **faire**
  $\beta_T(i) = 1$;
fin
pour *i= T-1 à 1* **faire**
  pour *j=1 à N* **faire**
    β_{t+1}(j)=$\sum_{i=1}^{N}(a_{ij}\beta_{t+1})b_j O_{t+1}$
  fin
fin

Algorithme de Viterbi

Le problème que l'on pose ici est de trouver, étant donnée une séquence de symboles $\{O\} = \{O_1, O_2, \ldots, O_T\}$ et un MMC λ= (A, B, π), la séquence d'états du MMC qui a la plus forte probabilité de générer O. Ce qui nous préoccupe n'est pas la valeur de la probabilité maximale

mais de trouver le chemin le plus probable appelé « chemin de Viterbi » noté q^* qui permet de générer la séquence $\{O\}$ avec cette probabilité. Cet algorithme est une solution du problème 2 évoqué dans la section précedente. On cherche le chemin le plus probable, on cherche donc le chemin $(q_1, q_2, \ldots, q_T)$ pour lequel la vraisemblance est maximale, c'est à dire la valeur maximale de $P(q_1, q_2, \ldots, q_T | O, \lambda)$. L'algorithme de Viterbi est un algorithme de programmation dynamique. Pour cela, on définit la variable $\sigma_t(i)$ la probabilité maximale des chemins se terminant à l'état i tel que :

$$\sigma_t(i) = \max_{q_1, q_2, \ldots, q_t} P[q_1, q_2, ..., q_t = i | O_1, O_2, ..., O_t, \lambda] \tag{2.15}$$

Par induction on obtient :

$$`\sigma_{t+1}(j) = \max_i [\sigma_t(i) a_{ij}] b_j(O_{t+1}) \tag{2.16}$$

Pour retrouver la séquence d'états, nous devons garder la trace de l'argument qui a maximisé 2.16, pour chaque t et j. Pour ce faire, nous utilisons le tableau $\psi_t(j)$. De la même manière que l'algorithme Forward et Backward, cette variable peut etre calculée d'une façon récursive :

1. Initialisation :

$$\sigma_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N \tag{2.17}$$

$$\psi_1(t) = 0 \tag{2.18}$$

2. Récursion :

$$\sigma_t(j) = \max_{1 \leq i \leq N} [\sigma_{t-1}(i) a_{ij}] b_j(O_t) \tag{2.19}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} \sigma_{t-1}(i) a_{ij} \quad \text{with} \quad 2 \leq t, 1 \leq j \leq N \tag{2.20}$$

3. Évaluation du meilleur score (i.e. chemin plus probable) :

$$P^* = \max_{1 \leq i \leq N} [\sigma_T(i)] \tag{2.21}$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\sigma_T(i)] \tag{2.22}$$

4. « Backtracking » des chemins :

$$q_T^* = \psi_{t+1}\left(q_{t+1}^*\right), t = T-1, T-2, ..., 1 \tag{2.23}$$

Mais dans la pratique, nous prenons le log de chacune des probabilités mises en jeu, ainsi, tous les produits de probabilité sont remplacés par des sommes. Nous pouvons proposer une version complète de l'algorithme Viterbi (algorithme 3) :

Algorithme 3 : Viterbi

Données : $\{O\} = \{O_1, O_2, ..., O_T\}$, λ=(A,B,π)
pour *i=1 à N* **faire**
  $\sigma_1(i) = \pi_i b_i(O_1)$ et $\psi_1(t) = 0$
fin
pour *t=2 à T* **faire**
  pour *j=1 à N* **faire**
    Calculer et stocker $\sigma_t(j)$ à l'aide de la formule 2.19
    Calculer et stocker $\psi_t(j)$ à l'aide de la formule 2.20
  fin
fin
Déterminer P^* et q_T^* à l'aide des formules 2.21 et 2.22
pour *t=T-1 à 1* **faire**
  Déterminer q_T^* à l'aide de la formule 2.23
fin

2.2.6 Apprentissage des MMC

Nous avons vu dans la section précédente les algorithmes classiques employés dans les MMC pour répondre aux problèmes de l'évaluation et le décodage. Dans cette partie, nous allons voir les algorithmes d'apprentissage des MMC. Nous pouvons distinguer dans le problème d'apprentissage des MMC deux cas de figure selon la structure (la topologie) connue ou pas. Dans notre cas, nous employons les MMC dans le cadre d'une classification supervisée donc c'est dans le cadre d'une structure connue. Dans la phase d'apprentissage, nous avons besoin de techniques qui permettent de capter l'information statistique sur la variabilité des formes représentant la même classe. L'intérêt des Modèles de Markov Cachés résulte de l'existence d'algorithmes efficaces permettant de déterminer la valeur optimale des paramètres des modèles. Partant d'une valeur initiale des paramètres, ces algorithmes itératifs alignent les modèles sur des observations connues l'ensemble d'apprentissage que nous le notons (O_{app}), et calculent les statistiques lors de ces productions. A partir de ces statistiques, ils déterminent un nouveau jeu de paramètres qui améliore la probabilité totale d'émission des observations de l'ensemble d'apprentissage. Cet ensemble doit posséder une taille importante pour que chaque phénomène particulier de variabilité soit suffisamment représenté. Les algorithmes d'apprentissage des MMC sont issus des algoritmes

d'Estimation/Maximization (EM) fondés sur un processus itiratif.

Estimation-Maximization (EM)

L'algorithme EM, proposé par Dempster et al [49] est une classe d'algorithme statistique qui s'applique principalement à l'estimation par maximum de vraisemblance dans le cadre de problèmes liés aux données incomplètes ou dans les modèles probabilistes lorsque le modèle dépend de variables non observables (états cachés) comme dans notre problématique des MMC. On l'appelle « algorithme EM » puisque chaque itération de l'algorithme consiste en une étape d'Estimation et une étape de Maximisation. L'algorithme EM tente de résoudre le problème suivant : sachant qu'un échantillon de la séquence $\{O\}$ est observé, mais que les Q correspondants ne sont pas observés (cachés). Le principe de cet algorithme est :

— **Étape E** : évaluation de l'espérance, où l'on calcule l'espérance de la vraisemblance en tenant compte des dernières variables observées ;

— **Étape M** : maximisation où l'on estime le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E.

On utilise ensuite les paramètres trouvés à l'étape M comme point de départ d'une nouvelle phase d'évaluation de l'espérance, etc. Dans les MMC, nous utilisons à partir d'un modèle initial, les formules de réestimation de Baum-Welch ou les meilleurs chemins de Viterbi permettent de réestimer le modèle. Ces itérations permettent de maximiser ma probabilité $P(\{O_{app}\}|\lambda)$ jusqu'à atteindre un critère de convergence (par exemple variation de $P(\{O_{app}\}|\lambda)$ inferieure à un seuil). Nous présentons dans ce qui suit les algorithmes de Baum-Welch et de Viterbi utilisés dans le cadre d'un apprentissage par maximum de vraisemblance.

Algorithme de Baum-Welch

L'algorithme de Baum-Welch est un algorithme de réestimation itératif, son but étant la maximisation de la vraisemblance d'un modèle λ. Celui-ci modifie substantiellement les paramètres du modèle étudié afin d'augmenter sa vraisemblance. L'algorithme réalise son optimisation en réestimant les différents paramètres (A, B et π), suivant la (ou les) séquence(s) observée(s). Les estimations peuvent donc logiquement se concevoir ainsi :

$$\overline{\pi}_i = probabilité\ d'être\ dans\ l'état\ S_i\ quand\ t = 1 \tag{2.24}$$

$$\overline{a}_{ij} = \frac{nombre\ de\ transitions\ de\ l'état\ S_i\ vers\ S_j}{nombre\ de\ fois\ où\ l'on\ quitte\ S_i} \tag{2.25}$$

$$\overline{b_j(k)} = \frac{nombre\ de\ fois\ où\ l'on\ est\ dans\ l'état\ S_j\ en\ observant\ le\ symbole\ v_k}{nombre\ de\ fois\ où\ l'on\ est\ dans\ l'état\ S_j} \tag{2.26}$$

Pour estimer ces nouvelles probabilités, l'algorithme de Baum-Welch utilise deux coefficients. Le coefficient $\xi_t(i,j)$, représente la probabilité d'être dans l'état S_i à l'instant t et de passer dans l'état S_j à l'instant t+1, d'après le modèle λ et la séquence d'observations $\{O_{app}\}$. Le coefficient $\gamma_t(i)$, représente la probabilité d'être dans l'état S_i à l'instant t, sachant l'observation $O_{app\ t}$ et le modèle λ. Le calcul de ces coefficients est rapidement réalisé grâce aux deux matrices de paramètres α et β, délivrées par les algorithmes Forward et Backward.

Calcul du coefficient ξ :

$$\xi_t(i,j) = \frac{\alpha_t(i).a_{ij}.b_j(O_{app\ t+1}).\beta_{t+1}(j)}{P(\{O_{app}\}|\lambda)} \tag{2.27}$$

Calcul du coefficient γ :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = \frac{\alpha_t(i).\beta_t(i)}{P(O_{app\ t}|\lambda)} \tag{2.28}$$

Ainsi, la réestimation des paramètres (A, B et π) du modèle Markovien λ est telle que :

$$\overline{\pi_i} = \gamma_1(i), 1 \leq i \leq N \tag{2.29}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, 1 \leq i,j \leq N \tag{2.30}$$

$$\overline{b_j}(k) = \frac{\sum_{t=1\ et\ O_t=v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}, 1 \leq i \leq N \tag{2.31}$$

Après avoir réestimé les différents paramètres du modèle d'origine, l'algorithme recalcule la vraisemblance du nouveau modèle. Il va ensuite réitérer les différentes opérations de réestimation avec ce nouveau modèle, tant que la condition de fin n'est pas atteinte. L'algorithme de Baum-Welch est presenté dans l'algorithme 4.

Nous pouvons prouver que la probabilité de génération de séquence augmente à chaque itération et que l'algorithme converge vers un optimum local [50]. Néanmoins, dans ce cas là, l'espace de recherche est continu et l'optimum local n'est en réalité jamais atteint [51].

Il est alors nécessaire de définir un critère d'arrêt, par exemple en stoppant la procédure lorsque

Algorithme 4 : *Baum – Welch*

Données : $(O_{app}) = \{\{O_{app\ 1}\}, \{O_{app\ 2}\}, ..., \{O_{app\ L}\}\}$, λ=(A,B,π)

pour *chaque séquence de la base de données d'apprentissage (l=1 à L)* **faire**
  Appliquer les algorithmes de Forward-Backward
  pour *t=1 à T* **faire**
    pour *i=1 à N* **faire**
      pour *j=1 à N* **faire**
        Calculer et stocker $\xi_t(i,j)$ à l'aide de la formule 2.27
      fin
      Calculer et stocker $\gamma_t(i)$ à l'aide de la formule 2.28
    fin
  fin
fin
Réestimer les paramètres λ= (A, B, π)
Tant que la condition de fin n'est pas atteinte

l'augmentation de $P(\{O_{app}\}|\lambda)$ est inferieure à un certain seuil, ou lorsque le nombre d'itérations est jugé suffisamment important.

Apprentissage par Viterbi

Comme dans l'apprentissage par Baum-Welch, l'apprentissage par Viterbi fonctionne sur le même principe c'est-à-dire un algorithme de réestimation itératif. Si les chemins de Viterbi des séquences d'apprentissage sont connus, alors chaque état, chaque transition et chaque symbole attaché aux états des MMC peuvent être associés au nombre de fois où ils sont utilisés pour générer ces séquences.

L'algorithme d'apprentissage de Viterbi est fondé sur le même principe que l'algorithme EM. Sauf que nous remplaçons la partie E par le chemin de Viterbi. Les paramètres ne sont donc pas estimés en maximisant la vraie probabilité de génération des séquences d'apprentissage, mais plutôt la probabilité de génération le long du chemin le plus probable. Soit $n_i, n_{i\to j}$ *et* n_i^O respectivement le nombre de fois où l'état i est utilisé, le nombre de fois où la transition $i \to j$ est utilisée et le nombre de fois où le symbole V_k est généré par l'état i dans le chemin de Viterbi :

$$\overline{a_{ij}} = \frac{n_{i\to j}}{n_i} \tag{2.32}$$

$$\overline{b_{ij}} = \frac{n_i^O}{n_i} \tag{2.33}$$

Finalement, l'algorithme de Viterbi consiste, à partir d'un paramétrage initial du MMC, à calculer les chemins les plus probables (chemin de Viterbi). Ces chemins (comme le montre la figure 2.5) sont utilisés par la suite pour calculer le nombre de fois où chaque transition, chaque état et chaque symbole attaché aux états est utilisé.

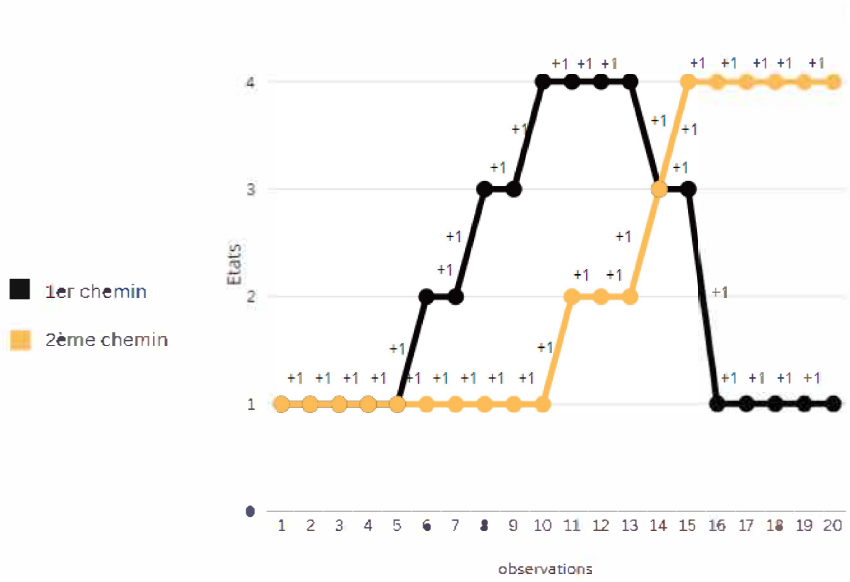

FIGURE 2.5 – Exemple de deux chemins les plus probables de Viterbi

A chaque fois que l'on passe d'un état à un autre, on incrémente de $+1$. La cellule de $n_{i \to j}$ du chemin de Viterbi est présentée dans le tableau 2.1.

De l'état i \ *Vers l'état j*	1	2	3	4
1	+4 +9	+1 +1	0	0
2	0	+1+2	+1 +1	0
3	1	0	+3	+1
4	0	0	+1	+3 + 5

TABLE 2.1 – Contribution des deux chemins de Viterbi

Ensuite, nous réestimons à l'aide des formules 2.32 et 2.33 les paramètres du MMC. Nous re-paramétrons la structure à l'aide de ces estimations et nous réitèrons jusqu'au critère de fin. L'algorithme d'apprentissage par Viterbi est détaillé dans algorithme 5.

Algorithme 5 : Apprentissage par Viterbi

Données : $(O_{app}) = \{\{O_{app\ 1}\},\{O_{app\ 2}\},...,\{O_{app\ L}\}\}$

Choisir un modèle initial $\lambda(A,B,\pi)$ Initialiser les variables : $n_i, n_{i\to j}$ *et* n_i^O

pour *chaque* $\{O_{app\ l}\}$ *dans* (O_{app}) **faire**

    Calculer le chemin Viterbi de $\{O_{app\ l}\}$

    Ajuster les variables $n_i, n_{i\to j}$ *et* n_i^O la contribution de $\{O_{app\ l}\}$

fin

Réestimer les paramètres $\lambda(A,B,\pi)$ en utilisant les formules 2.32, 2.33

Jusqu'au critère de fin

Nous pouvons montrer que la valeur de $P((O_{app})|\lambda)$ augmente à chaque itération et que l'algorithme converge vers un optimum local [52]. Malheureusement, il existe généralement un grand nombre d'optimaux locaux et le paramétrage obtenu par ce processus dépend fortemenet du paramétrage initial choisi.

2.2.7 Comparaison entre Baum-Welch et Viterbi

L'algorithme de Viterbi ne prend en compte que les événements correspondant à l'alignement optimal entre les formes observables et le modèle. L'algorithme de Baum-Welch prend en considération tous les alignements possibles et maximise la vraie probabilité d'émission des observations. Toutefois, en classification, les deux algorithmes fournissent sensiblement des performances identiques même si l'estimation des probabilités par l'algorithme de Viterbi est moins bonne. L'algorithme de Viterbi converge plus rapidement que l'algorithme de Baum-Welch, car, il ne tient compte que de la meilleure séquence d'états qui a pu produire la séquence d'observations. Il est également plus rapide car il nécessite moins de calculs. Cet algorithme nécessite beaucoup plus de données pour converger vers une « bonne » solution [53].

2.3 Intérêt des Modèles de Markov Cachés

Les MMC tirent leur intérêt de plusieurs propriétés que nous présentons dans les points suivants :

— Les MMC possèdent une représentation complémentaire dans leurs deux ensembles de paramètres. Les probabilités d'observation de symboles représentent le phénomène mor-

phologique. Elles ont la capacité d'approximer toute fonction de densité de probabilité arbitraire. L'autre ensemble de paramètres, les probabilités de transition, représente l'intéraction temporelle (liée à l'ordre) entre les composantes d'une forme. Finalement, l'optimisation conjointe de ces deux ensembles de paramètres fait que les MMC sont particulièrement adaptés à la modélisation de signaux variant dans le temps.

— Quelque soit le domaine, le problème de reconnaissance reste difficile du fait de l'existence de plusieurs sources de variabilités, contexte, environnement etc. Pour traiter ce problème de manière efficace, nous avons besoin d'une très grande quantité d'exemples d'apprentissage pour une modélisation correcte des phénomènes observés. L'ajustement manuel entre les formes et les classes associées, dans une telle base, est alors une tâche extrêmement fastidieuse, voire impossible. Les MMC, eux, possèdent une telle capacité, en posant d'abord des hypothèses sur la structure du modèle qui nécessite une bonne connaissance du système, et en utilisant ensuite la probabilité que les modèles aient généré la base d'apprentissage (critère du maximum de vraisemblance) pour améliorer les estimations de paramètres.

— L'apprentissage des paramètres à l'aide des MMC et des techniques de réestimation possède plusieurs propriétés recherchées dans les systèmes de reconnaissance :

  — Il nécessite une supervision minimale,

  — Il s'appuie sur une base mathématique garantissant la convergence vers un point critique optimal,

  — Il s'accommode convenablement de l'augmentation du nombre d'exemples d'apprentissage, ne nécessitant qu'une augmentation linéaire des calculs,

  — La reconnaissance de formes très variables telles que la parole ou l'écriture requiert le raisonnement avec des incertitudes. Les MMC étant des modèles probabilistes, ils sont des candidats naturels à la représentation de ces formes,

  — La reconnaissance implique une recherche dans un espace d'états pour une solution optimale ou sous optimale. Les recherches basées sur les MMC diffèrent des approches ascendantes qui propagent les erreurs et ne peuvent intégrer des connaissances descendantes, et des approches descendantes qui sont souvent très complexes à mettre en oeuvre. En intégrant hiéarchiquement plusieurs niveaux de représentation, la recherche par MMC est globale, où toutes les sources de connaissance participent à

chaque décision. Finalement, l'utilisation de probabilités nous permet d'avoir un mécanisme d'évaluation des solutions cohérent.

2.4 Limitation des MMC

L'utilisation des MMC pour des problèmes réels rencontre quelques difficultés que l'on peut résumer de la manière suivante :

— Les MMC s'appuient sur l'hypothèse d'indépendance des probabilités pour déterminer la probabilité d'observation d'une séquence de symboles. Cela découle de l'hypothèse que les observations extraites d'un signal sont indépendantes entre elles, et que la génération d'un symbole d'observation n'est fonction que de l'état courant de la chaîne de Markov. Il est clair que pour des signaux réels, cette hypothèse est abusive et ne peut qu'approximer la réalité. Toutefois, elle a l'avantage de simplifier les calculs.

— Dans [54], Levinson et al ont étudié plusieurs exemples concrets pour juger du comportement des techniques de modélisation par MMC sur des données appropriées, et notamment vérifier que les algorithmes d'estimation de paramètres sont corrects. Nous pouvons dire que même sous des conditions idéales (c.à.d lorsque les données sont générées par un MMC connu a priori et que donc nous avons l'avantage de connaître le vrai modèle), les modèles calculés en apprentissage peuvent contenir des artefacts et peuvent ne pas représenter de manière fidèle la structure inhérente des données traitées. Par conséquent, une grande attention et des validations empiriques sont requises pour l'utilisation de ces techniques.

Malgré ces inconvénients, les MMC sont bien adaptés à la reconnaissance de signaux tels que la parole [55] [56] ou l'écriture [57] où ils ont obtenu d'excellents résultats dans des applications relatives à ces domaines.

2.5 Problèmatique multi-classe

Dans le domaine de l'apprentissage automatique, la classification est le problème de l'identification de l'appartenance d'une nouvelle instance à un ensemble de catégories (classes). Lorsque les catégories contiennent plus de deux étiquettes différentes, on parle de classification multi-classe. Au cours des dernières années, une question qui a reçu beaucoup d'attention dans les problèmes

de classification est celle de savoir comment obtenir de meilleurs taux de classification. Ce problème devient encore plus complexe lorsque le nombre de classes est élevé. Dans ce scénario multi-classe (voir figure 2.6), on suppose que les étiquettes des classes sont indépendantes les unes des autres, ce qui explique que la plupart des techniques et méthodes proposées pour améliorer les performances des classificateurs s'appuient sur ce principe.

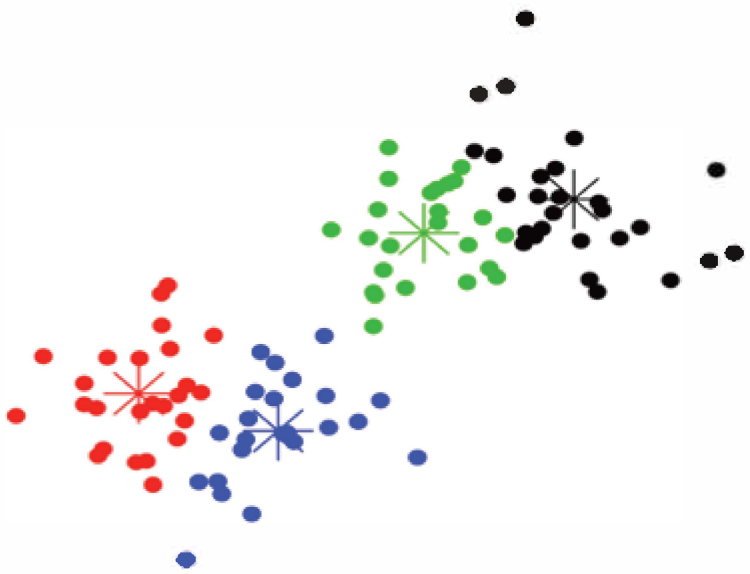

FIGURE 2.6 – Problème des multi-classes

L'un des principaux objectifs de la résolution d'une tâche de classification est de faire des prédictions aussi "bonnes" que possible, la notion de "bonnes" dépendant de la mesure d'évaluation utilisée pour évaluer la qualité du classifieur. Par exemple, en matière de classification, la précision est l'une des mesures d'évaluation les plus populaires. Cependant, il n'est pas toujours facile d'obtenir de bonnes prédictions.

Et les choses sont généralement encore plus compliquées dans le cas d'une classification multiple, car le classifieur doit alors faire la distinction entre un grand nombre de classes pour effectuer les prédictions. C'est pourquoi, au cours des dernières décennies, de nombreux efforts ont été déployés pour améliorer les performances des classifieurs. La plupart des approches proposées dans la littérature relèvent de l'un de ces trois groupes :

— Concevoir de meilleures techniques d'apprentissage,

— Appliquer une sorte de transformation sur les données d'apprentissage,

— Modifier ou ajuster les prédictions données par les classifieurs.

Dans notre cas, nous nous intéressons au premier groupe c'est-à-dire concevoir de meilleures techniques d'apprentissage. En effet, l'apprentissage automatique et la science du comportement, se sont généralement concentrés sur les comportements qui se sont produits ou se produiront [58]. Ces comportements sont appelés comportements positifs (PB) ou comportements occurrents mais, peu de recherches ont été menées sur l'analyse, la détection ou la prévision des comportements non récurrents (CNR), c'est-à-dire ceux qui ne se sont pas produits ou ne se produiront pas. Les comportements non récurrents sont également appelés comportements négatifs. Ils ne sont pas simples, car ils sont généralement cachés et difficiles à comprendre, ou bien on ne s'en préoccupe généralement pas.

Le fait que les CNR soient négligés ne signifie pas qu'ils sont sans importance. Cependant, l'absence de certains éléments dans les séquences peut également être intéressante dans certains scénarios. La figure 2.7 illustre le principe de comportement et non-comportement.

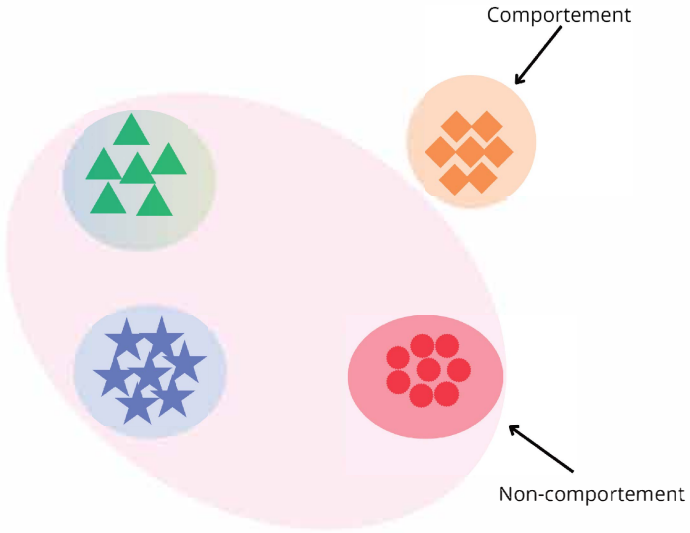

FIGURE 2.7 – Principe de comportement et non-comportement

2.6 Conclusion

Nous avons, dans ce chapitre, donné un état de l'art sur les MMC. Nous avons vu la relation entre les MMC et la maintenance conditionnelle. La partie suivante fut consacrée à la présentation des différents algorithmes utiles à la bonne exploitation des MMC. Dans ce chapitre, nous avons

abordé la problématique des multi-classes. C'est dans ce cadre que notre recherche se positionne, en effet, deux problématiques ont été posées :

— Est-ce que ces non-comportements contiennent de l'information ?

— Est-ce que l'information contenue dans ces non-comportements est complémentaire aux informations contenues dans le comportement ?

Dans les chapitres suivants, nous allons vérifier ces deux problématiques à travers des protocoles et des tests effectués en utilisant les MMC comme outil de classification.

Chapitre 3

Méthode de de classification par les Modèles de Markov Cachés

3.1 Introduction

Dans le chapitre précédent, nous avons fait un état de l'art sur les MMC. Nous allons maintenant nous focaliser sur les modes de fonctionnement d'un système en utilisant des méthodes de classification.

Comme nous l'avons indiqué dans les chapitres précédents, notre travail se positionne dans le cadre de la maintenance conditionnelle en utilisant la méthode de classification. Le but est d'identifier une classe parmi plusieurs possibles. Ces classes sont les différents modes de défaillance possibles auxquels s'ajouterait le mode sans défaillance.

Durant la phase d'apprentissage, il faut disposer de données étiquetées pour chacune des classes. Par exemple, sur la figure 3.1, nous illustrons une situation avec quatre classes pour lesquelles quatre modèles doivent être estimés.

L'approche que nous proposons consiste à modéliser le système en effectuant l'apprentissage de chacune de ces classes par un MMC. Le but est de classer un échantillon de mesure du corpus de test (O^r_{test}) dans l'une des classes représentées dans la figure 3.1.

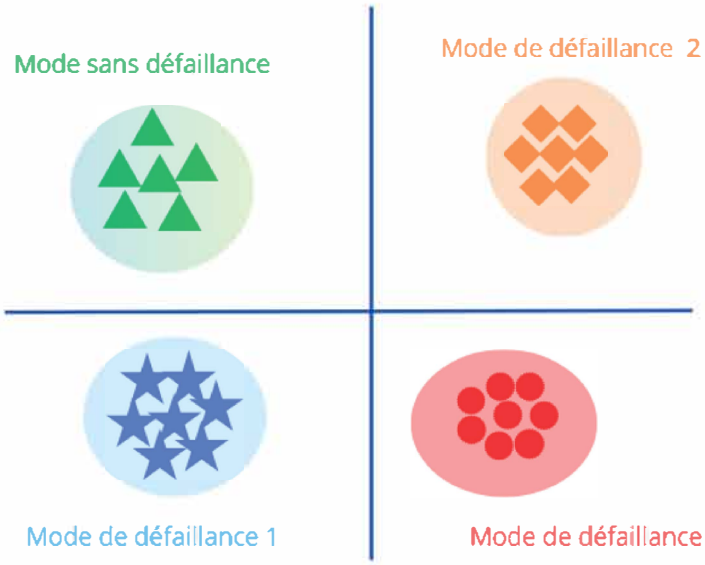

FIGURE 3.1 – Exemple avec 4 classes (3 modes de défaillance plus un mode sans défaillance)

Ce chapitre sera constitué comme suit :

Dans un premier temps, nous introduisons les modèles de synthèse utilisés pour cette étude, afin d'obtenir des données similaires à celles d'un processus industriel.

Ces modèles établissent ainsi des relations cohérentes entre les observations de sortie d'un processus industriel et l'état « caché » de celui-ci. Ensuite, nous étudierons la ressemblance entre les modèles de synthèse générés grâce aux mesures de ressemblance. Nous expliquerons comment nous séparons les données issues des modèles de synthèse en corpus d'apprentissage et de test. Nous étudierons l'homogénéité entre les deux corpus à l'aide du principe de l'entropie de Shannon ainsi que l'information mutuelle. Nous expliquerons la méthode proposée ainsi que les résultats obtenus. Enfin, nous ferons varier le nombre de séquences de la base d'apprentissage afin d'étudier l'influence de la quantité de données sur les résultats de classification.

3.2 Modèles de synthèse

Dans cette partie, nous présentons notre démarche pour produire des données qui permettront de tester la méthode d'apprentissage proposée. Une même topologie est utilisée pour chaque classe pour produire des données associées à cette classe. Ces données sont ensuite réparties en deux corpus : l'un pour l'apprentissage, l'autre pour les tests. Pour construire un modèle de synthèse, il nous faut définir :

— La topologie,

— La matrice de transition A,

— La matrice de distribution des observations B,

— Et enfin un vecteur d'initialisation π.

Afin de s'assurer de la qualité des données, plusieurs tests statistiques seront effectués.

3.2.1 Un modèle de synthèse par classe

L'objectif de notre travail est d'effectuer le diagnostic à travers les méthodes de classification. Dans ce cas, le but est de classer un échantillon de mesure (observation) dans l'une des classes qui représente un état de santé d'un système industriel (voir la figure 3.1). Pour cela, nous avons créé un modèle de synthèse pour chaque classe. La première étape de création d'un modèle de synthèse est de choisir une topologie. La topologie d'un MMC dépend directement des éléments non nuls de la matrice de transition notée A. En particulier, un MMC dont la matrice A est pleine (aucun terme nul) est dite à connectivité ou topologie ergodique : toutes les possibilités d'évolution sont permises. En pratique, il s'avère que la matrice A est très rarement pleine car elle fixe la topologie et l'orientation future de l'analyse du modèle choisi. C'est pour cette raison que nous avons choisi la topologie gauche-droite déjà utilisée pour l'étude d'un système industriel [59]. A partir d'une topologie de type gauche-droite (figure 3.2), d'un nombre d'états N notés de S_1 à S_N, et d'un nombre M de symboles, plusieurs jeux de paramètres de modèles λ_r= (A_r, B_r, π_r) avec r $\in [1,R]$ R le nombre de classes, sont choisis aléatoirement avec Matlab.

A partir de la toolbox Matlab, les modèles de synthèse λ_r permettent de produire des $2-uplets$ (symboles, états) de manière aléatoire en respectant la topologie choisie, la topologie gauche-droite. Pour la phase de qualification des méthodes d'apprentissage, seules les observations sont utilisées. Cependant, pour qualifier la qualité de ces données, nous disposons également de la correspondance état/observation.

Nombre de classes

Pour les tests effectués, nous avons choisi 4 classes (trois modes de défaillance et un mode sans défaillance) ($R = 4$).

Nombre d'états

Nous avons choisi d'utiliser une topologie à 4 états notés S1, S2, S3 et S4.

Nombre de symboles

Les symboles correspondent aux observations liées au système à modéliser i.e. ce que l'on veut ou que l'on peut observer. Il faut que ces observations soient significatives a priori. Nous avons choisi ici un alphabet de 16 symboles (les symboles émis représentent par exemple des interventions de maintenance).

Topologie de la chaîne de Markov

Une topologie gauche-droite a été choisie comme le montre la figure 3.2.

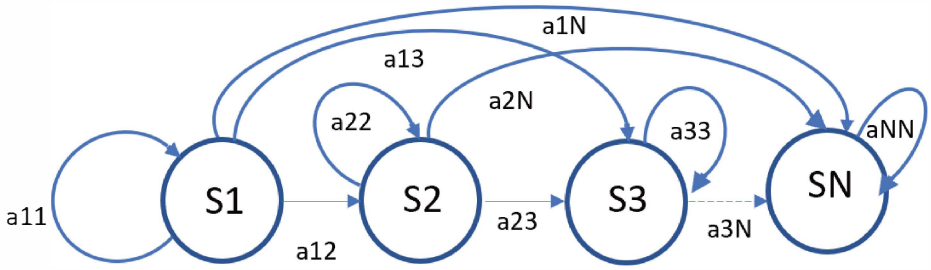

FIGURE 3.2 – Topologie de type gauche-droite

Avec ces paramètres, des simulations ont permis de produire des séquences de longueur T=20 pour chacune des classes. La figure 3.3 illustre la répartition des données pour chaque classe.

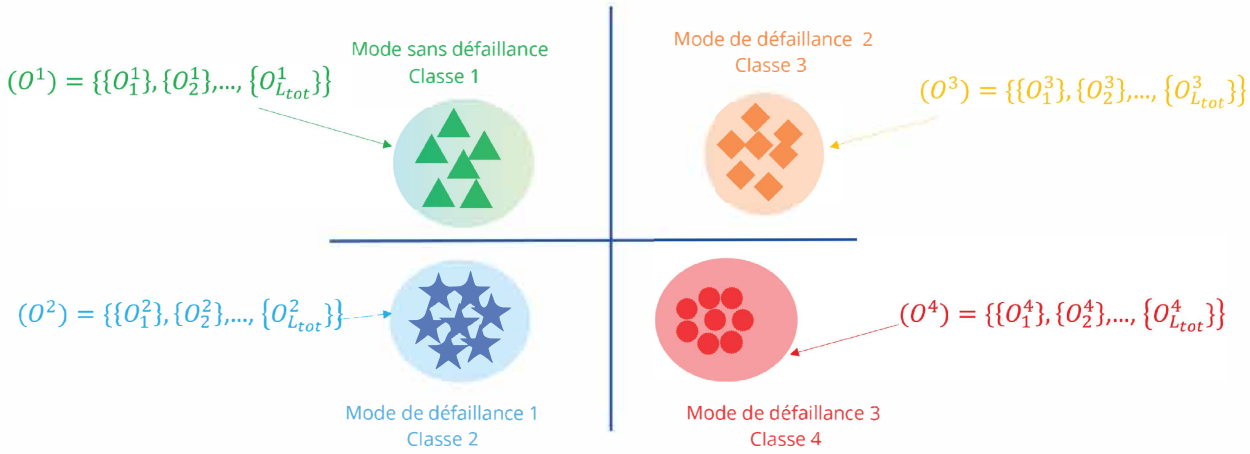

FIGURE 3.3 – Production des bases de données

Nous avons lancé une première simulation afin de produire $L_{tot} = 1000$ observations. La matrice A_1 (équation 3.1) de taille N x N est un exemple de matrice de transition.

Cette matrice est générée de façon aléatoire à partir d'une topologie imposée (gauche-droite).

Elle permet de produire les données de synthèse qui forment les corpus d'apprentissage et de test.

$$A_1 = \begin{pmatrix} 0.0680 & 0.2303 & 0.3191 & 0.3826 \\ 0 & 0.3841 & 0.6022 & 0.0137 \\ 0 & 0 & 0.9612 & 0.0388 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.1}$$

La matrice de distribution des observations B_1 de taille N x M d'un exemple de modèle de synthèse sera de la forme suivante :

$$B_1 = \begin{pmatrix} 0.0985 & 0.1149 & \dots & \dots & \dots & \dots & 0.0250 & 0.0735 \\ 0.0872 & 0.0080 & \dots & \dots & \dots & \dots & 0.0833 & 0.0169 \\ 0.0897 & 0.0092 & \dots & \dots & \dots & \dots & 0.0248 & 0.0809 \\ 0.0630 & 0.0052 & \dots & \dots & \dots & \dots & 0.1206 & 0.0180 \end{pmatrix} \tag{3.2}$$

Le vecteur d'initialisation π_1 sera de la forme :

$$\pi_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.3}$$

3.2.2 Répartition des données en corpus d'apprentissage et de test

Une fois les données générées, nous les répartissons en deux corpus : apprentissage et test. Nous avons choisi l'approche Train-Test Split qui consiste à décomposer de manière aléatoire un ensemble de données. Une partie servira à l'apprentissage du modèle, l'autre partie permettra de le tester pour la validation. En général, on réserve 70 à 80% des données pour l'apprentissage. Les 20 à 30% restants seront exploités pour la Cross-Validation (données test). Les figures 3.4, 3.5, 3.6 et 3.7 montrent la répartition des symboles pour les deux corpus (apprentissage et test).

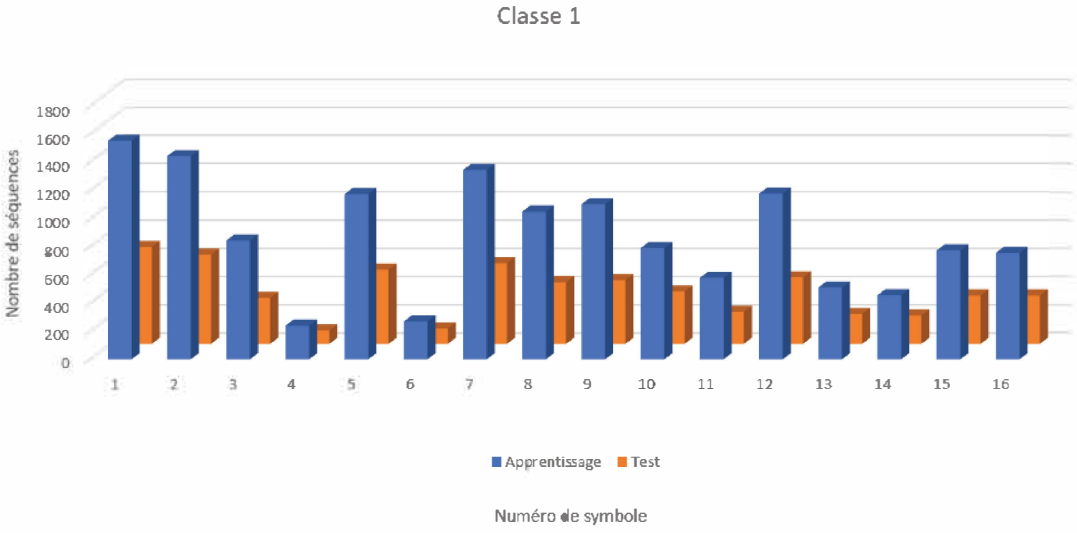

FIGURE 3.4 – Distribution des symboles pour les corpus apprentissage et test de la classe 1

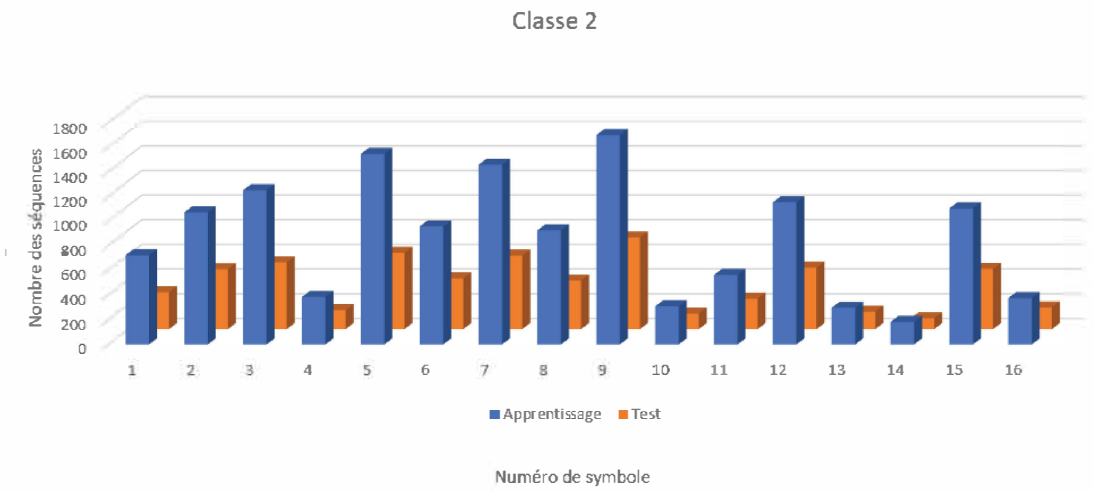

FIGURE 3.5 – Distribution des symboles pour les corpus apprentissage et test de la classe 2

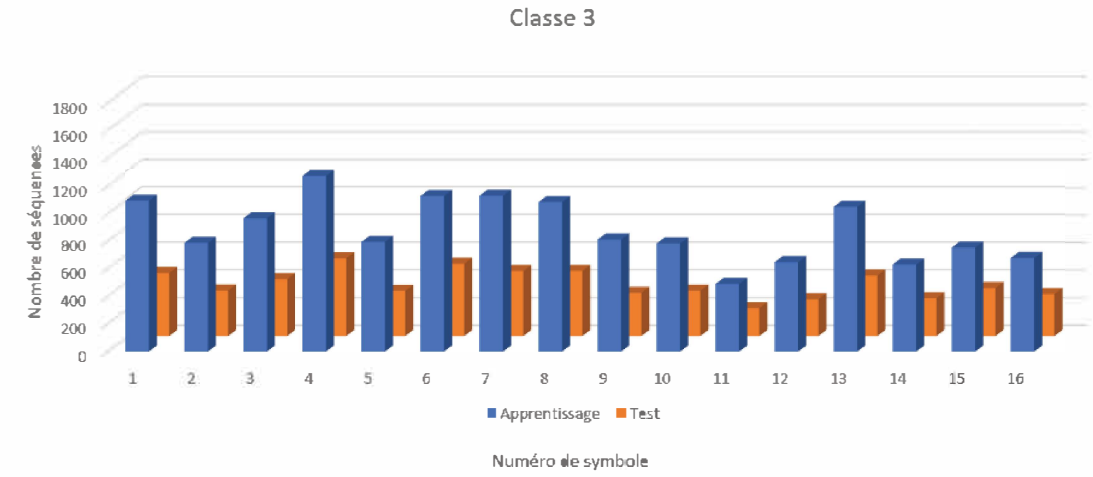

FIGURE 3.6 – Distribution des symboles pour les corpus apprentissage et test de la classe 3

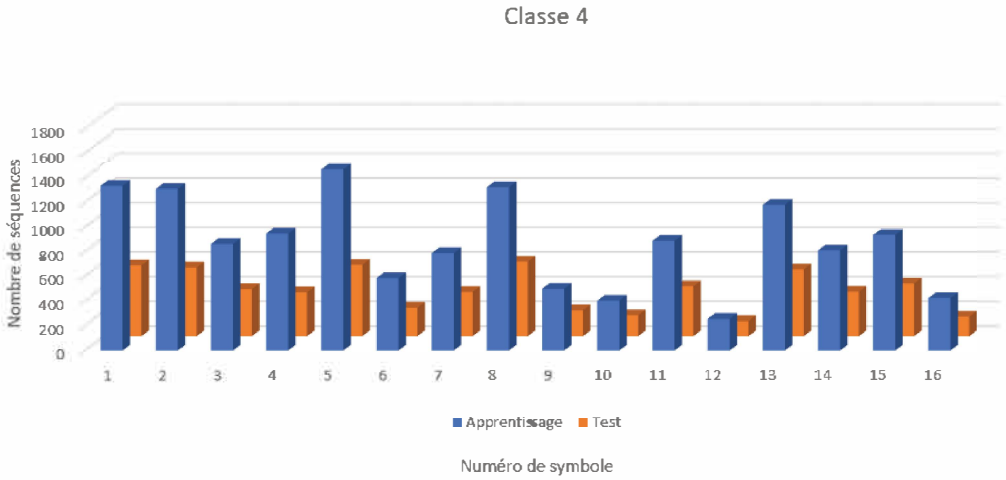

FIGURE 3.7 – Distribution des symboles pour les corpus apprentissage et test de la classe 4

D'après ces figures, nous remarquons que la répartition des symboles pour les corpus d'apprentissage et de test est homogène pour chaque classe. Ces figures montrent également que la répartition des symboles est différente d'une classe à l'autre.

3.3 Classification par Modèles de Markov Cachés

La classification se fonde sur les observations ainsi que sur leurs classes d'origine pour identifier les classes des séquences d'observations inconnues. En d'autres termes, les algorithmes de classification devraient prévoir l'étiquette de chaque nouvelle observation. Dans notre cas, nous nous sommes orientés vers une approche probabiliste pour faire cette classification. Cette méthode est fondée sur les Modèles de Markov Cachés que l'on peut utiliser dans le diagnostic des défauts. La classification permet de définir une fonction d'appartenance à partir d'un ensemble de classes constitué d'échantillons d'apprentissage. Chaque échantillon est constitué d'entrée (observation) et d'une valeur de sortie désirée (étiquette de la classe). La fonction d'appartenance doit alors estimer une classe d'appartenance correcte pour n'importe quel echantillon. Il faut pour cela effectuer les étapes suivantes :

— Construire un ensemble d'apprentissage. L'ensemble d'apprentissage doit être représentatif de l'utilisation réelle de la fonction d'appartenance. Ainsi, un ensemble d'echantillons est collecté et les sorties correspondantes sont également réunies, soit par une expertise humaine ou à partir des mesures réalisées.

— Déterminer la structure de la fonction d'appartenance et de l'algorithme d'apprentissage

correspondant. Dans notre cas, nous avons choisi l'algorithme de Baum–Welch ainsi que l'algorithme d'apprentissage par Viterbi.

— Évaluer la précision de la fonction d'appartenance. Après ajustement des paramètres de la fonction d'appartenance par un algorithme d'apprentissage, la performance de la fonction résultante doit être mesurée sur un ensemble de test qui est séparé de l'ensemble d'apprentissage.

3.4 Méthode proposée

Dans cette partie nous expliquons les différentes étapes de notre méthode :

— **Étape 1 :** *Sélection aléatoire des modèles*

La première étape consiste à générer λ_r modèles avec Matlab. Ces modèles représentent R classes différentes.

— **Étape 2 :** *Génération de données*

Des observations synthétiques sont générées à partir de ces R modèles, en définissant un nombre L_{tot} de séquences d'observations $(O) = \{O_1, O_2, ..., O_{L_{tot}}\}$, et une longueur T.

— **Étape 3** : *Corpus d'apprentissage et de test*

Ces séquences d'observations sont divisées en deux corpus : les données d'apprentissage et les données de test. Nous avons opté pour 2/3 des données utilisées pour l'apprentissage du modèle et 1/3 pour les données de test (voir les équation ci-dessous).

$$(O_{app}) = \{O_{app\ 1}\}, \{O_{app\ 2}\}, ..., \{O_{app\ L}\} \tag{3.4}$$

$$(O_{test}) = \{O_{test\ 1}\}, \{O_{test\ 2}\}, ..., \{O_{test\ L-L_{tot}}\} \tag{3.5}$$

— **Étape 4** : *phase d'apprentissage*

Pour la phase d'apprentissage, nous avons utilisé les algorithmes de Viterbi et de Baum-Welch pour la réestimation de chaque modèle λ_r.

— **Étape 5** : *Phase d'identification et de classification*

Pour vérifier la capacité des modèles à identifier chaque classe, nous avons utilisé le maximum de vraisemblance.

Les cinq étapes sont présentées sur la figure 3.8. Dans la section suivante, nous allons détailler la phase d'apprentissage ainsi que la phase de test (classification ou identification).

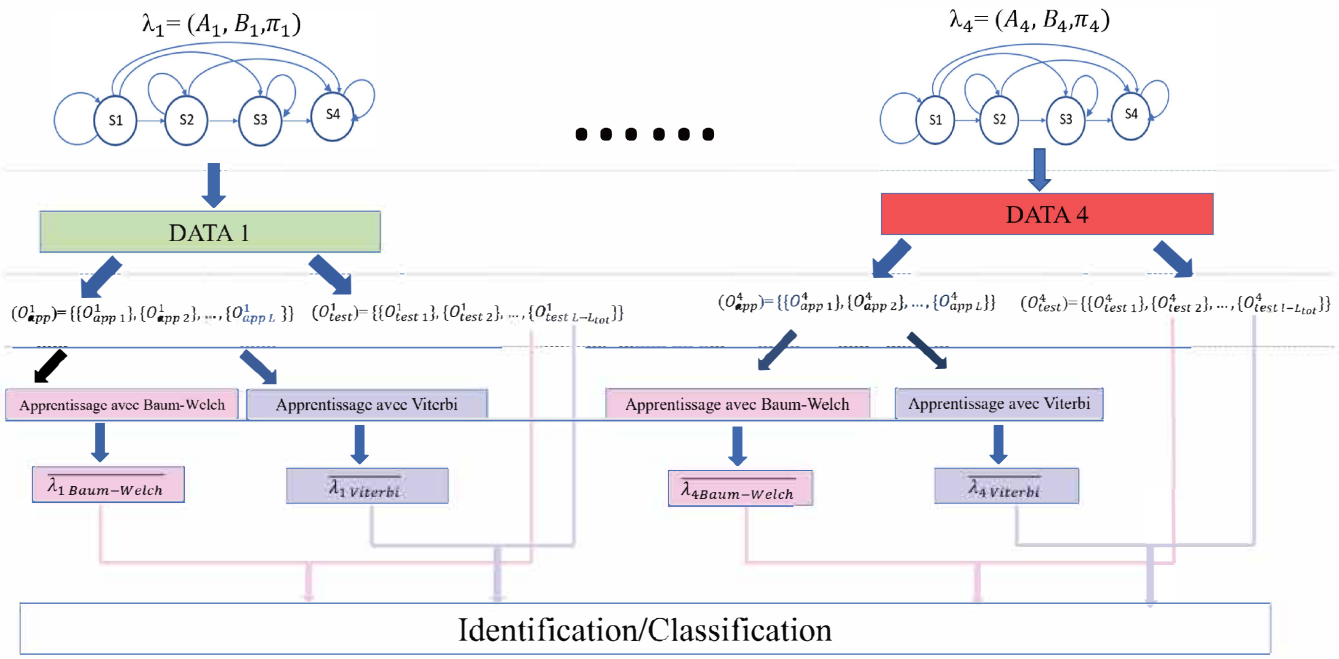

FIGURE 3.8 – Les étapes de classification par MMC avec données de synthèse

3.4.1 Phase d'apprentissage

Dans la phase d'apprentissage, nous avons besoin de techniques qui permettent de capter l'information statistique sur la variabilité des formes. L'intérêt des Modèles de Markov Cachés résulte de l'existence d'algorithmes efficaces permettant de déterminer la valeur optimale des paramètres des modèles. Partant d'un jeu de valeurs initiales des paramètres, ces algorithmes itératifs alignent les modèles sur des observations connues (l'ensemble d'apprentissage), et calculent les statistiques lors de ces productions. A partir de ces statistiques, ils déterminent une nouvelle valeur des paramètres qui améliore la probabilité totale d'émission des observations de l'ensemble d'apprentissage. Cet ensemble doit posséder une taille importante pour que chaque phénomène particulier de variabilité soit suffisamment représenté.

Dans notre cas, la phase d'apprentissage des modèles consiste à réestimer les paramètres de chaque modèle λ_r. Les seules informations disponibles sur le système sont ses différents modes de fonctionnement, autrement dit ses classes. Ainsi, l'approche que nous proposons consiste à

modéliser le système en effectuant l'apprentissage de chacune de ses classes par un MMC. La phase d'apprentissage consiste à utiliser les corpus apprentissage de chaque classe (O^r_{app}) qui vont nous servir pour ajuster les paramètres de chaque MMC de façon à maximiser la probabilité de générer chaque séquence d'observations $\{O^r_{app}\}_l = \{O^r_{app\ 1}, O^r_{app\ 2}, \ldots, O^r_{app\ T}\}$ étant donné un modèle initial λ. La réponse au problème d'apprentissage est apportée par les algorithmes d'apprentissage de Baum-Welch et de Viterbi comme nous l'avons expliqué dans le chapitre 2.

3.4.2 Phase de test

Etant donné un corpus d'observations inconnue (O^r_{test}) de chaque classe et un Modèle de Markov Caché. Nous avons besoin de calculer les probabilités $P(\{O^r_{test}\}|\bar{\lambda}_r)$, afin de déterminer le modèle le plus ressemblant (classe d'origine) à cette séquence. Pour la classification, nous calculons le maximum de vraisemblance pour comparer plusieurs modèles et faire l'identification. Si le modèle r est le plus probable, il faut déclarer que la classe de la séquence est la classe r.

3.4.3 Résultats de classification

Nous présentons dans cette section les résultats obtenus par la classification par des Modèles de Markov Cachés sur les bases de données test appelés (O^r_{test}). Pour cela, nous avons généré quatre modèles λ_r avec r $\in$[1,R] ici R=4.

- La topologie de chaque modèle est de type gauche-droite à quatre états (N = 4) avec émission de 16 symboles différents.
- Chaque base de données (O^r) contient L_{tot} = 1000 séquences d'observations.
- Les corpus d'apprentissage (O^r_{app}) contiennent 700 séquences.
- Les corpus de test (O^r_{test}) contiennent $L_{tot} - L$ = 300 séquences d'observations.
- La longueur de chaque séquence est de T = 20.

Dans un premier temps, nous avons effectué l'apprentissage à l'aide de l'algorithme Baum-Welch. Nous avons calculé la vraisemblance de chaque séquence de test avec chacun des modèles $P(\{O^r_{test}\}_l|\bar{\lambda}_r)$, pour l'identification, nous avons cherché le maximum de vraisemblance. Pour éviter de juger une situation qui pourrait être anecdotique, liée au tirage aléatoire des paramètres des modèles, les tests ont été répétés plusieurs fois. La même démarche a été faite en remplaçant

l'algorithme d'apprentissage Baum-Welch par l'algorithme d'apprentissage de Viterbi.

Nous montrons dans le tableau 3.1 la matrice de confusion obtenue après apprentissage par l'algorithme de Baum-Welch, pour un jeu de données (premier tirage des paramètres).

	classe 1	classe 2	classe 3	classe 4	la somme en %
classe 1	95	3.33	0.33	1.33	100%
classe 2	14.66	72.66	8.66	4	100%
classe 3	1.33	1.33	93.66	3.66	100%
classe 4	2.33	2	2.33	93.33	100%

TABLE 3.1 – Matrice de confusion du premier tirage avec apprentissage Baum Welch

Pour ce jeu de données, on peut noter que les erreurs commises sont presque equitablement réparties. Nous remaquons que la classe 2 est la plus proche de la classe 1 avec un taux de reconnaissance de 14.66% Le pourcentage global obtenu est de 88.66% (la moyenne de la diagonale). Ce chiffre correspond au tirage 1 de la figure 3.9.

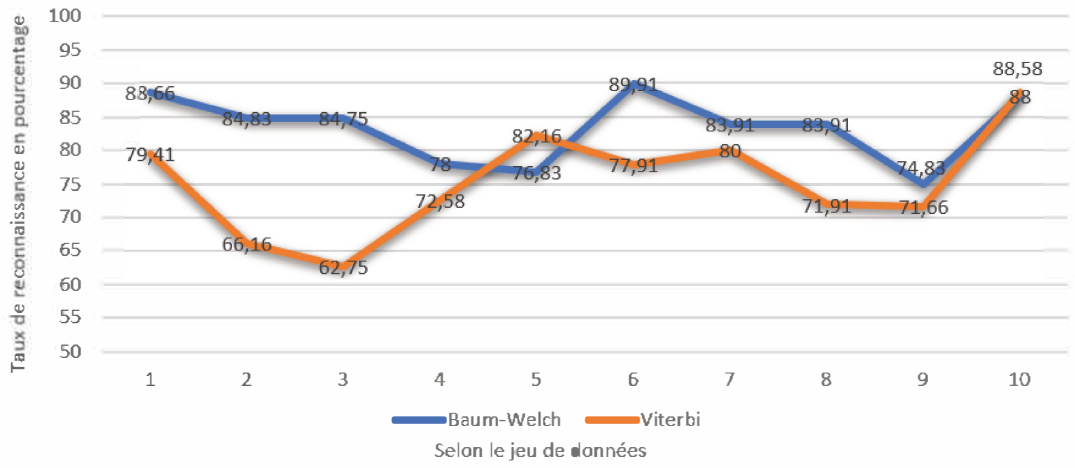

FIGURE 3.9 – Résultat de classification en utilisant Baum-Welch et Viterbi

Le tableau 3.2 montre la matrice de confusion pour les mêmes données après un apprentissage effectué par l'algorithme de Viterbi.

	classe 1	classe 2	classe 3	classe 4
classe 1	91	6.33	1.33	1.33
classe 2	13.66	72.33	10.33	3.66
classe 3	2.33	2.66	91.66	3.33
classe 4	11	10.66	15.66	62.66

TABLE 3.2 – Matrice de confusion du premier tirage avec apprentissage Viterbi

Les résultats obtenus sont inferieurs à ceux obtenus avec Baum-Welch mais les erreurs sont similaires.

3.5 Confiance dans les réponses de classification

Afin de juger la qualité de la réponse apportée par la classification (écart entre les vraisemblances n°=1 , n°=2, n°=3, n°=4), nous avons tracé ces vraisemblances normalisées en fonction du numéro des séquences. Pour cela, nous avons normalisé les vraisemblances par rapport à la vraisemblance de la classe étiquetée.

Dans la pratique, nous avons calculé les log-vraisemblances, donc la normalisation s'effectue par une simple différence comme le montre l'équation 3.6.

$$log(\frac{A}{B}) = log(A) - log(B) \tag{3.6}$$

La valeur de réference sera donc zéro pour la classe étiquetée. Les valeurs de vraisemblance normalisée supérieure à zéro indiqueront une erreur de classification.
Les valeurs trés inferieures à zéro indiqueront un grand écart entre la bonne réponse apportée et les vraisemblances des autres classes.
Les figures 3.10, 3.11, 3.12 et 3.13 montrent l'écart entre les vraisemblances normalisées pour chaque classe du premier tirage avec algorithme d'apprentissage de Viterbi. Nous avons choisit les 150 premières séquences à identifier. L'intervalle a été limité de -2 à 10.

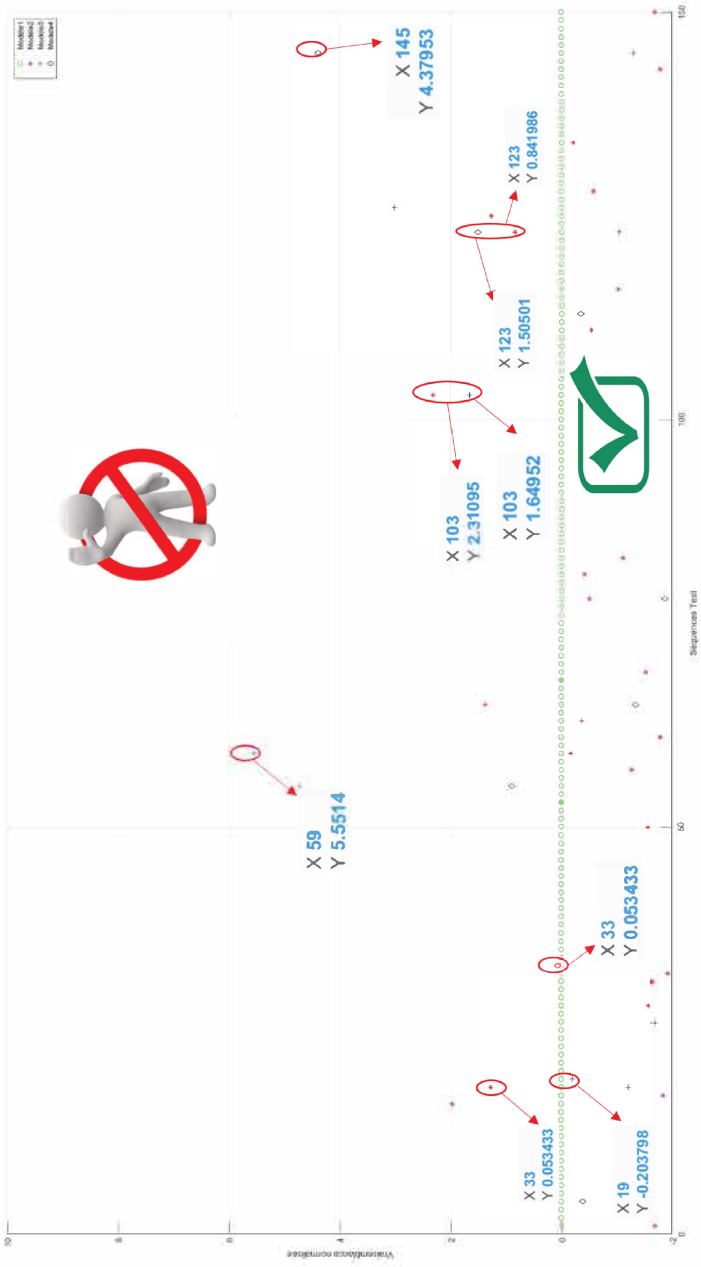

FIGURE 3.10 – Ecart entre la classe 1 et les autres classes

Sur la figure 3.10 qui représente l'écart entre la classe 1 et les autres classes, nous remarquons que pour la séquence 33, le Modèle le plus proche est le Modèle 4 avec une vraisemblance normalisée de 0.05. Pour cette séquence, une erreur est commise avec un ecart très petit. Tandis que sur la séquence 123, nous remarquons que la classe identifiée est la classe 4 avec une meilleure vraisemblance et en 2ème position c'est la classe 2. Nous remarquons aussi que sur 150 séquences à identifier le nombre d'erreur est très faible ce qui reflète le taux de reconnaissance de 91%.

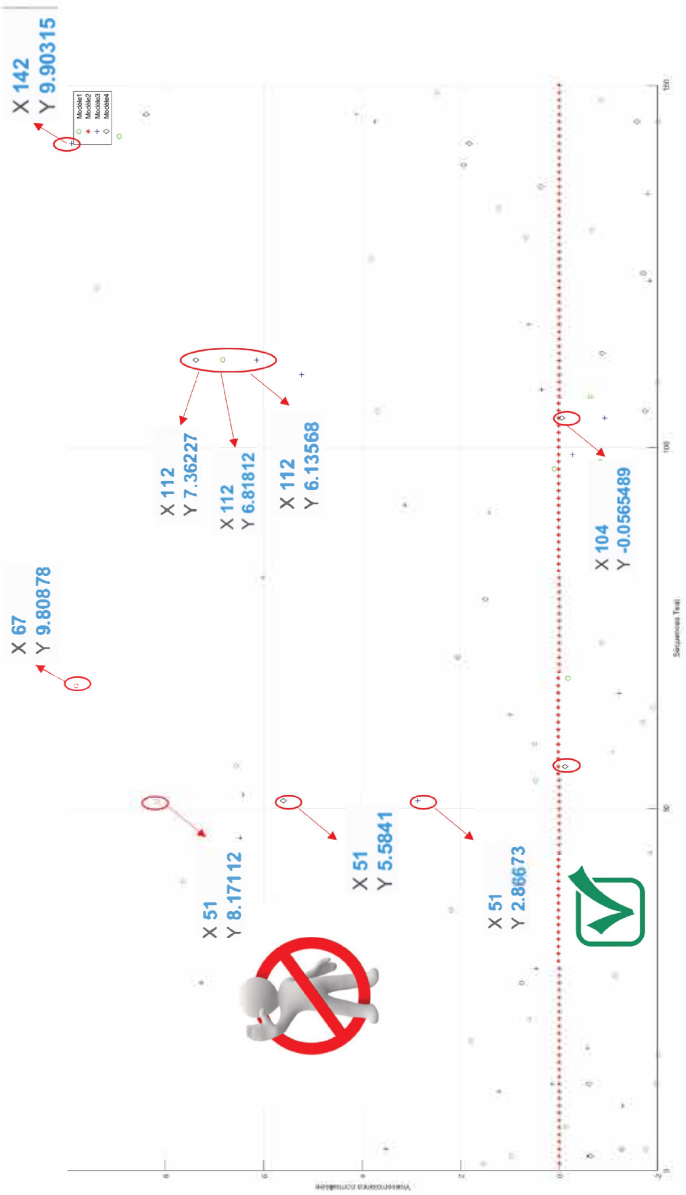

FIGURE 3.11 – Ecart entre la classe 2 et les autres classes

Sur la figure 3.11 qui représente l'écart entre la classe 2 et les autres classes, nous remarquons que pour la séquence 142, la classe identifiée est la classe 4. Pour cette séquence, une erreur est commise avec un écart très grand. Pour les séquences 51 et 112, le Modèle 2 identifie les 3 autres classes au lieu de la classe 2. Nous remarquons aussi que sur 150 séquences à identifier, de nombreux points sont au-dessus de zéro ce qui reflète le taux de reconnaissance global de 72.33% dans le tableau 3.2.

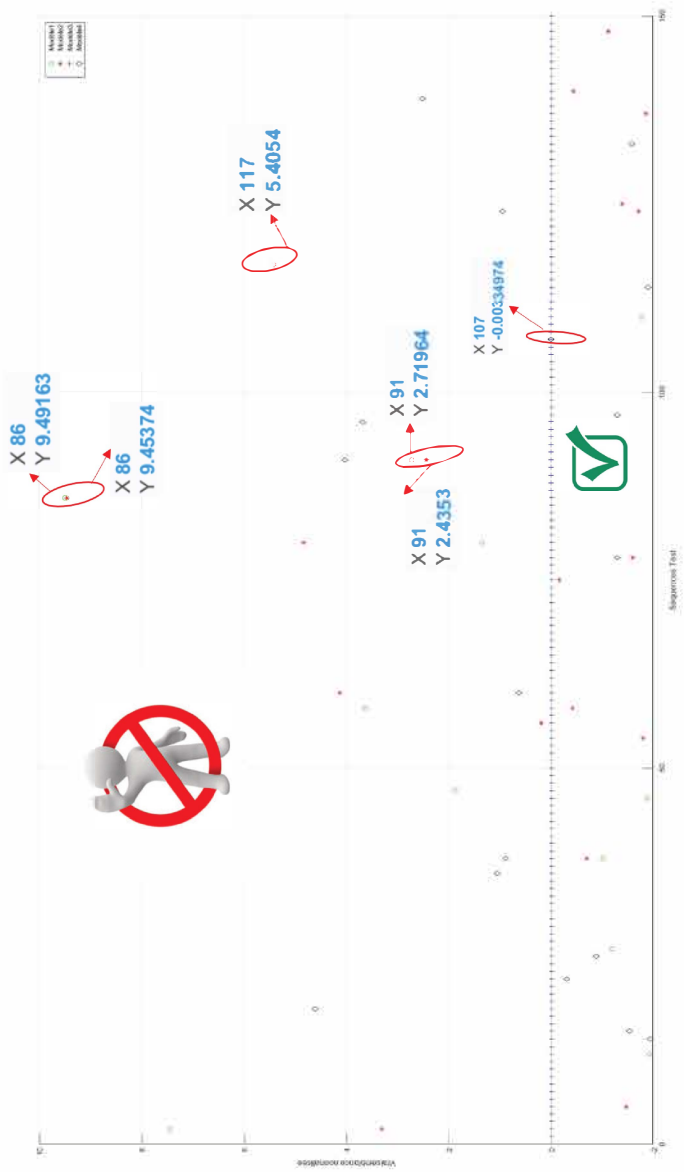

FIGURE 3.12 – Ecart entre la classe 3 et les autres classes

Sur la figure 3.12 qui représente l'écart entre la classe 3 et les autres classes, nous remarquons que pour la séquence 47, la classe identifiée est la classe 1 avec une meilleure vraisemblance. Nous remarquons aussi que sur 150 séquences à identifier, l'erreur est très faible ce qui reflète le taux de reconnaissance de 91.66%.

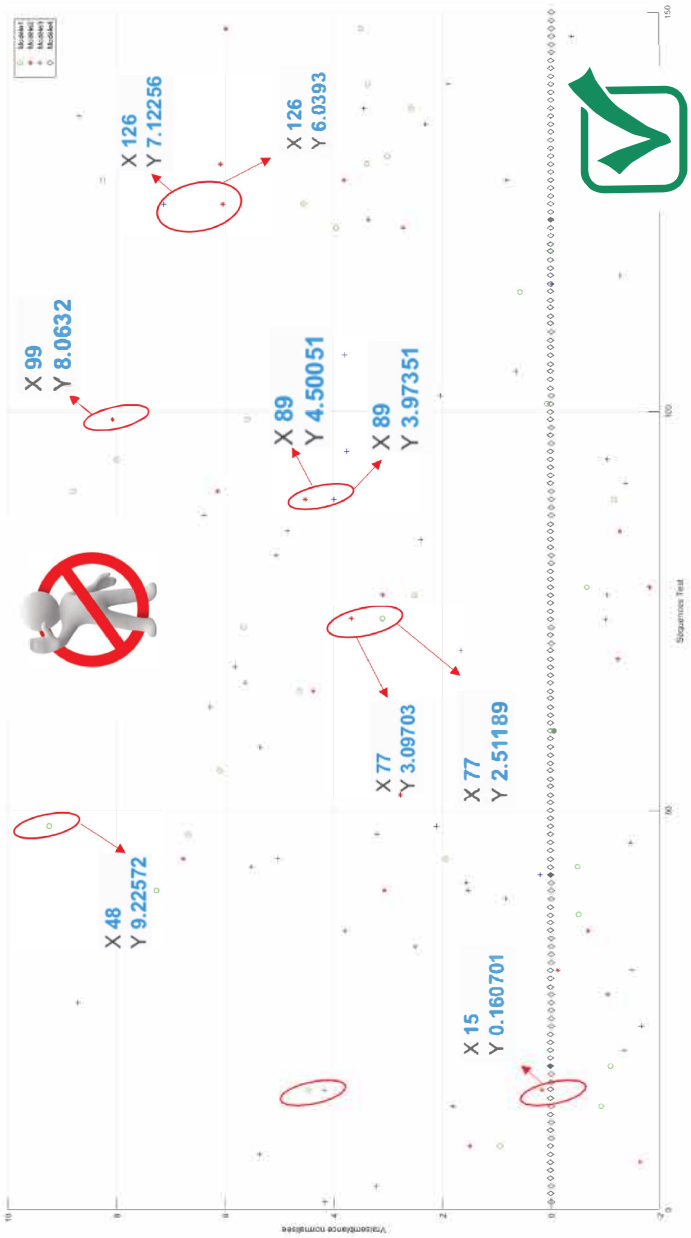

FIGURE 3.13 – Ecart entre la classe 4 et les autres classes

Enfin sur la figure 3.13 qui représente l'écart entre la classe 4 et les autres classes, nous remarquons que pour la séquence 77, la classe identifiée est la classe 2 et en 2ème position c'est la classe 1. Nous remarquons aussi que sur 150 séquences à identifier, le nombre d'erreur est très important avec un taux de reconnaissance de 62.66%.

3.6 Vérification de la pertinence des données

Dans cette partie, nous allons utiliser les principes de la théorie de l'information développés par E. Shannon [60]. Ces principes décrivent les aspects les plus fondamentaux des systèmes. Cette théorie s'intéresse à la construction et à l'étude de modèles mathématiques à l'aide essentiellement de la théorie des probabilités. Parmis les mesures fondamentales de la Théorie de l'information, nous trouvons l'entropie et l'information mutuelle. Dans la section suivante, nous allons définir ces deux mesures.

3.6.1 Mesure d'entropie

Selon Bouquet [61], l'entropie de Shannon est une fonction mathématique qui permet de mesurer la quantité d'information contenue dans une source. Cette source peut être un texte écrit dans une langue donnée, un signal électrique ou encore un fichier informatique quelconque, etc. Wolsztynsk [62] propose de minimiser l'entropie des variables aléatoires de ces modèles, dans le cas discret, afin d'y estimer les événements rares.

Définition de l'entropie de Shannon

L'entropie de Shannon est définie dans [63] et [64] comme suit :

$$H(s) = -\sum_{i=1}^{n} P_i log P_i \tag{3.7}$$

Avec P_i la probabilité moyenne de voir apparaître le symbole i dans S.

Propriétés formelles de l'entropie de Shannon : deux propriétés nous intéressent dans la théorie de l'entropie :

- **la minimalité :** une distribution « pure » (un seul symbole) a une entropie nulle : si un seul symbole est représenté, l'impureté est nulle. On a donc une première condition de minimalité : l'entropie d'une distribution est nulle si cette distribution est pure.
- **la maximalité** : à l'inverse, une distribution mélangée est « impure » : la valeur de l'entropie, lorsque plusieurs symboles sont représentés, doit donc être élevée. C'est la condition de maximalité : l'entropie d'une distribution est maximale lorsque tous les symboles sont représentés dans les mêmes proportions.

Pour de plus amples informations, le lecteur se réfèrera à Beirlant et al [65] qui présentent un état de l'art concernant les méthodes d'estimation de l'entropie et leurs propriétés.

Afin de vérifier la bonne répartition des données dans le corpus d'apprentissage et de test, nous avons utilisé l'entropie de Shannon.

Toutes les séquences ont été considérées comme "inconnues". Nous souhaitons savoir si la répartition des données est homogène.

L'entropie a été calculée sur chaque corpus et pour chacun des différents jeux de données de synthèse. La figure 3.14 montre des valeurs d'entropie proche pour chacun des corpus d'apprentissage et de test.

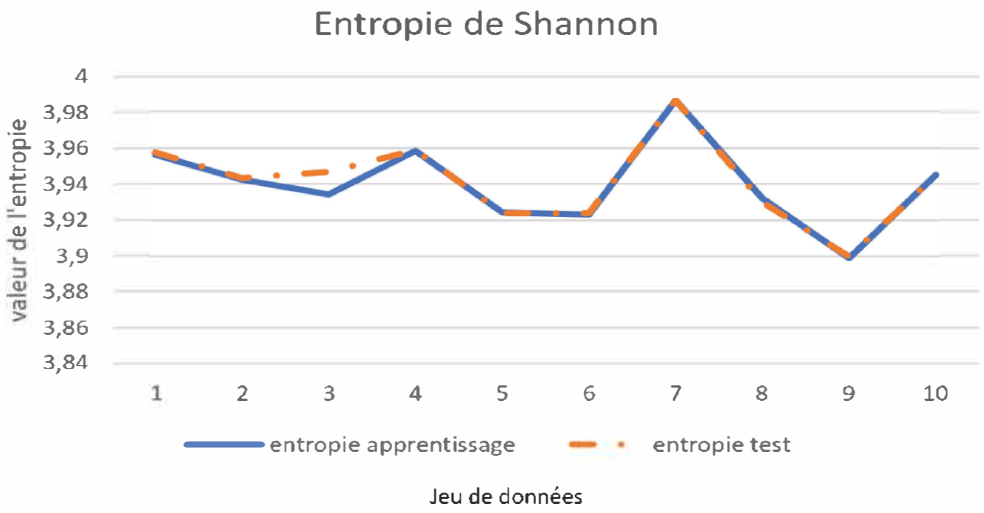

FIGURE 3.14 – Résultats d'entropie pour dix tirages pour les corpus d'apprentissage et de test

Cette figure montre une bonne répartition des séquences entre les deux corpus. Elle montre également une fluctuation de l'entropie en fonction du jeu de données.

3.6.2 Information mutuelle

Nous avons calculé l'information mutuelle dans chaque corpus.

Le calcul de l'information mutuelle est décrit dans l'équation 3.8 avec X qui represente les symboles et Y les classes.

Dans le cas discret :

$$I(X,Y) = \sum_{ij} P(X=i, Y=j) log \frac{P(X=i, Y=j)}{P(X=i)P(Y=j)} \tag{3.8}$$

Propriété 3.1.3.1 : I(X, Y) = 0 si X et Y sont des variables aléatoires indépendantes.

Comme dans la présentation des résultats de l'entropie, nous avons tracé les courbes de l'information mutuelle du corpus d'apprentissage et de test sur 10 tirages aléatoires. Le résultat est présenté dans la figure 3.15.

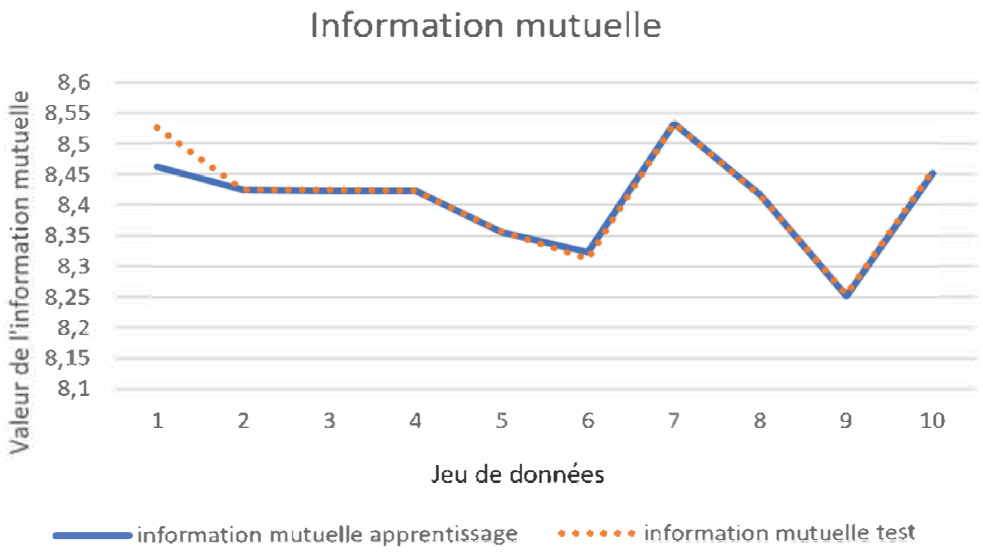

FIGURE 3.15 – Information mutuelle des dix tirages pour les corpus d'apprentissage et de test

Comme pour l'entropie, nous remarquons que l'information mutuelle du corpus apprentissage est presque identique à l'entropie du corpus test. Ces résultats montrent que la quantité d'information de chaque corpus est identique. On peut donc également conclure sur une bonne répartition des séquences dans les deux corpus.

3.6.3 Entropie et information mutuelle vs taux de reconnaissance

Nous avons tracé, sur la figure 3.16, pour chacun des jeux de données, le taux de reconnaissance obtenu avec un apprentissage par Baum-Welch ainsi que la valeur d'entropie calculée sur la base de données correspondante.

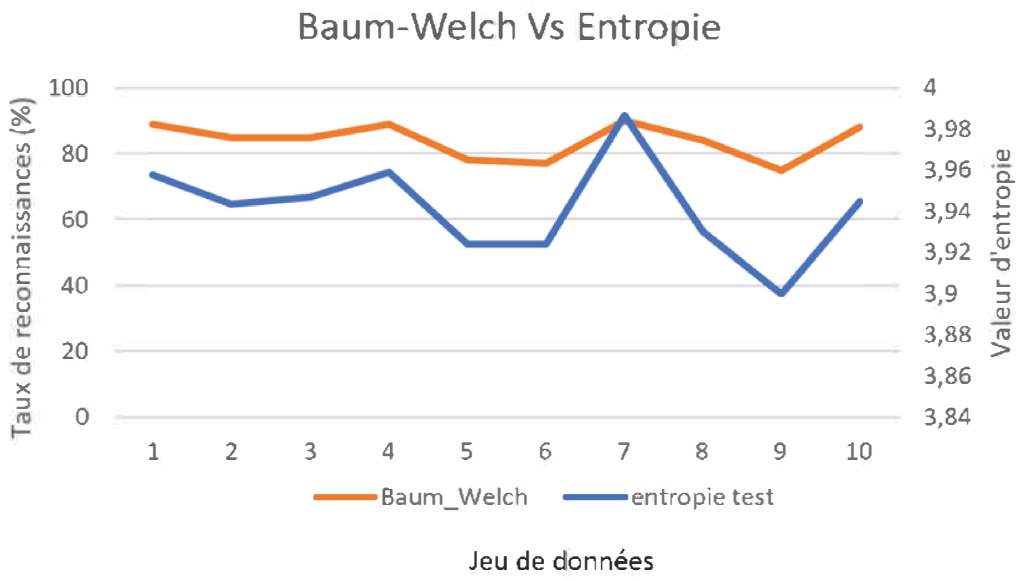

FIGURE 3.16 – Comparaison entre l'entropie et le taux de reconnaissance avec apprentissage Baum Welch

De même sur la figure 3.17, nous copmarons la valeur de l'information mutuelle avec les taux de reconnaissance obtenus.

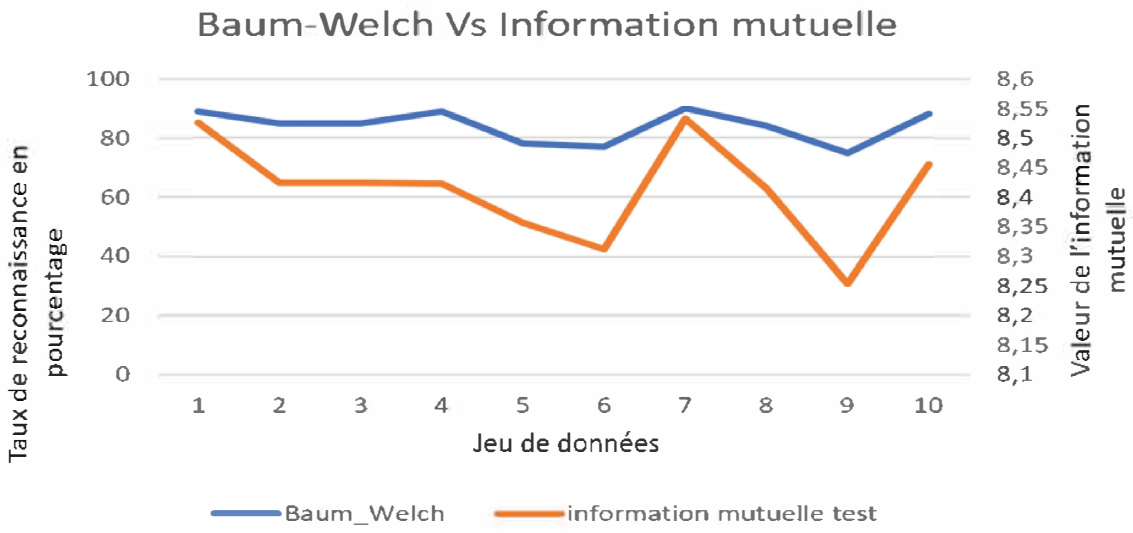

FIGURE 3.17 – Comparaison entre information mutuelle et taux de reconnaissance avec apprentissage Baum-Welch

Nous remarquons, sur les deux figures, la forte correlation entre l'entropie et le taux de reconnaissance (de même avec l'information mutuelle).

Dans cette étude, afin de faciliter la discussion sur nos résultats, il est préférable de choisir des jeux de données avec une entropie (ou information mutuelle) plus importante.

3.6.4 Évolution de l'entropie selon la taille du corpus de données

Sur la figure 3.18, nous avons tracé l'évolution de l'entropie en fonction du nombre de séquences dans la base de données pour différents jeux de données.

Nous constatons que les entropies convergent vers un maximum au bout de 150 séquences environ.

L'apport de nouvelles séquences au niveau de l'entropie, est minime au dela de 200 séquences.

Comme dans les travaux de B. Robles [66], nous pouvons donc estimer qu'une taille de base de données superieure au seuil estimé précédemment sera suffisante pour traduire les "variations" de chaque classe.

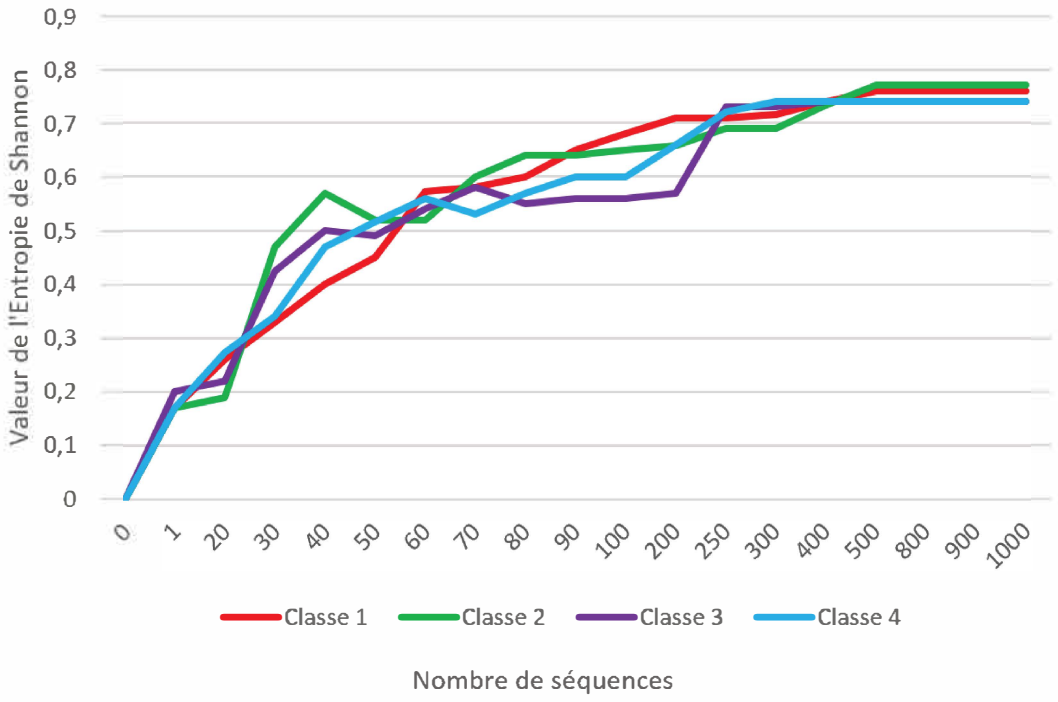

FIGURE 3.18 – Evolution de l'Entropie de Shannon selon la taille du corpus de données

3.7 Conclusion

Dans ce chapitre, nous avons expliqué notre approche, qui permet de classer les différents modes de défaillance d'un système industriel. Nous avons ainsi abordé la classification usuelle en utilisant les Modèles de Markov Cachés. Nous avons introduit la notion de modèle de simulation que nous appelons modèle de synthèse. Nous avons montré le comportement stochastique de ce modèle de synthèse. Celui ci nous a permis de générer des séquences d'observations parfaitement défini entre niveau de dégradation et symboles observés.
Nous avons décomposé nos attentes théoriques comme suit : étudier la pertinence des observations ou symboles en utilisant les techniques de la théorie de l'information (entropie et information mutuelle).

Nous avons montré comment nous séparons les données issues du modèle de synthèse en deux corpus d'apprentissage et de test et l'homogénéité entre ces deux corpus. Enfin, nous avons présenté les résultats issus de la classification usuelle, et l'influence de la quantité d'information sur les résultats de la reconnaissance.
Dans le chapitre suivant, nous allons aborder notre contribution afin d'améliorer l'apprentissage des MMC.
Les résultats de classification, présentés dans ce chapitre, serviront de références pour montrer l'apport de notre contribution.

Chapitre 4

Modèles Vs Non-Modèles

4.1 Introduction

Les deux chapitres précédents ont permis de présenter les Modèles de Markov Cachés : les différentes topologies, les differents algorithmes pour l'evaluation, le décodage ainsi que l'apprentissage. Nous avons aussi expliqué la méthode de classification usuelle en utilisant les MMC avec les algorithmes d'apprentissage de Baum-Welch et de Viterbi.

C'est dans ce cadre que notre travail se positionne. En effet, cette thèse a pour objectif de contribuer à l'amélioration de l'apprentissage des Modèles de Markov Cachés. Comme nous l'avons abordé dans le chapitre 2, le but de ce travail est de répondre aux problématiques multi-classes. Les MMC ont été utilisés dans des problèmes multi-classe en les combinant avec d'autres classifieurs, comme dans [67], pour effectuer une reconnaissance vocale multi-classes fondée sur un SVM avec une segmentation guidée par un MMC, ou [68] qui ont fusionné un réseau de neurones profond et un MMC pour reconnaître le comportement multi-classe anormal des personnes âgées. Les MMC ont également été utilisés dans les problèmes de clustering, comme dans le travail de Smyth [69].

Dans notre étude, l'objectif n'est pas de faire du clustering à l'aide de MMC, mais d'améliorer l'apprentissage des MMC afin de résoudre les problèmes multi-classe. Pour ce faire, nous introduirons une nouvelle stratégie d'apprentissage. Cette stratégie s'inspire des principes de non-comportement (NBA)[58].

Ce chapitre sera organisé comme suit :

Dans un premier temps, nous allons définir les termes Modèle et Non-modèle, ainsi que la nouvelle stratégie d'apprentissage. Nous montrerons que les Non-modèles contiennent de l'information. Finalement, nous évoquerons un nouvel algorithme d'apprentissage permettant d'améliorer l'apprentissage des MMC.

4.2 Définition du terme Non-modèle

Commme nous l'avons abordé précedemment, ce travail se positionne dans le cadre d'application visant à identifier une classe parmi plusieurs possibles. Ces classes sont, par exemple, les différents modes de défaillance possibles auxquels s'ajouterait le mode sans défaillance (comme nous l'avons expliqué dans le chapitre 2). Durant la phase d'apprentissage, il faut disposer de données étiquetées pour chacune des classes. Avec les méthodes d'apprentissage classiques citées dans le chapitre 2, chaque Modèle est estimé en utilisant les échantillons propres à sa classe. Dans notre étude, nous souhaitons intégrer une nouvelle stratégie d'apprentissage. Cette stratégie consiste à utiliser les informations potentiellement contenues dans les autres classes dans ce que nous avons nommé les Non-modèles. Comme le montre la figure 4.1, le pendant du Modèle 1 serait l'association des échantillons des autres classes pour représenter le Non-modèle 1.

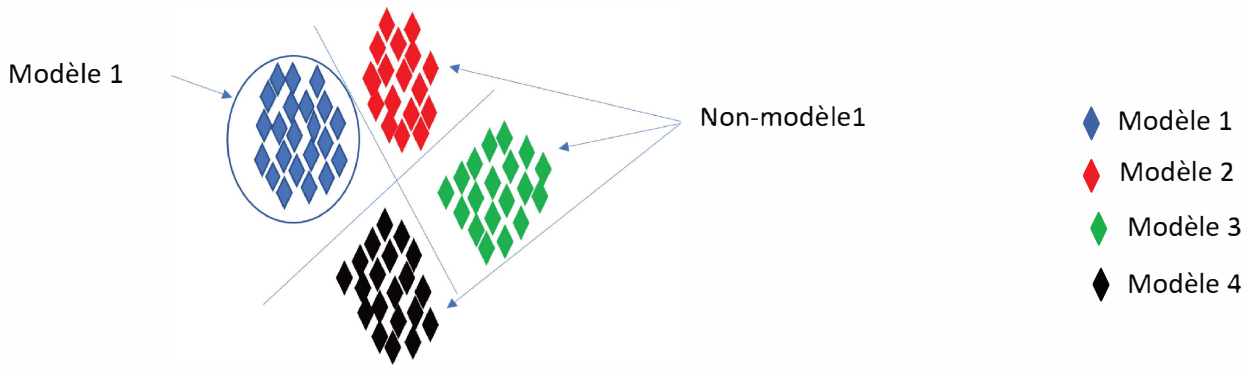

FIGURE 4.1 – Principe des Modèles et Non-modèles

Jusqu'à présent, le problème n'a pas été bien posé, et très peu de méthodes ont été proposées pour exploiter les informations contenus dans les Non-modèles. Or, ils sont considérés comme un outil très important pour l'informatique comportementale [70]. En effet, les Non-modèles (tels que l'absence de traitements médicaux) sont essentiels et parfois beaucoup plus informatifs que

les modèles (par exemple, l'utilisation d'un service médical) dans de nombreux systèmes et applications intelligents tels que les systèmes de transport intelligents, les soins de santé et la gestion des risques, car ils impliquent souvent des comportements non récurrents, mais intéressants.

L'extraction des informations contenues dans les Non-modèles a suscité un intérêt croissant dans les recherches récentes sur l'exploration de données [71], car elle prend en compte les relations négatives entre les éléments, qui sont ignorées par l'extraction de motifs Modèle. Les chercheurs dans [72] explorent une méthode efficace pour l'extraction des informations contenues dans les Non-modèles afin de modéliser les événements et les comportements fréquents et non fréquents. L'extraction des informations contenues dans les Non-modèles est efficace pour la modélisation difficile des non-occurrences d'événements et de comportements et de leurs combinaisons avec des événements occurrents.

Les méthodes existantes sont fondées sur l'incorporation de diverses contraintes dans les représentations de Non-modèles. Les Modèles dans les bases de données transactionnelles des clients sont couramment exploités pour les recommandations en matière de commerce électronique. Dans de nombreuses applications pratiques, l'absence de certains ensembles d'articles et de séquences peut avoir des conséquences importantes. L'extraction de séquences fréquentes comprenant non seulement l'occurrence mais aussi l'absence de certaines séquences augmentera la précision des recommandations de produits. Un modèle séquentiel contenant au moins un ensemble d'éléments absent est appelé Non-modèle. Les auteurs dans [73] ont formulé le problème de l'extraction des informations contenues dans les Non-modèles en introduisant des contraintes pratiques. Ils ont proposé un algorithme appelé : Positive and Negative Sequential Pattern Mining (PNSP) pour l'extraction. Les motifs découverts peuvent alors être plus intéressants et plus efficaces. Dans un premier temps, notre objectif est de vérifier que les Non-modèles ont la capacité à discriminer les classes (reconnaître ce qu'ils sont en apprenant ce qu'ils ne sont pas).

Dans la section suivante, nous définissons notre méthode qui consiste à effectuer l'apprentissage en utilisant à la fois les Modèles et Non-modèles.

4.3 Discrimination par Non-modèle

Dans cette section, nous allons expliquer la méthode d'apprentissage des MMC en utilisant les Non-modèles. Pour cela, nous devons constituer les corpus d'apprentissage de chaque Non-modèle. En effet, chaque Modèle a un pendant qui consiste à combiner les échantillons des autres classes (le contraire de la forme) afin de former les Non-modèles. La formation des corpus d'apprentissage du Non-modèle est illustrée dans la figure 4.2.

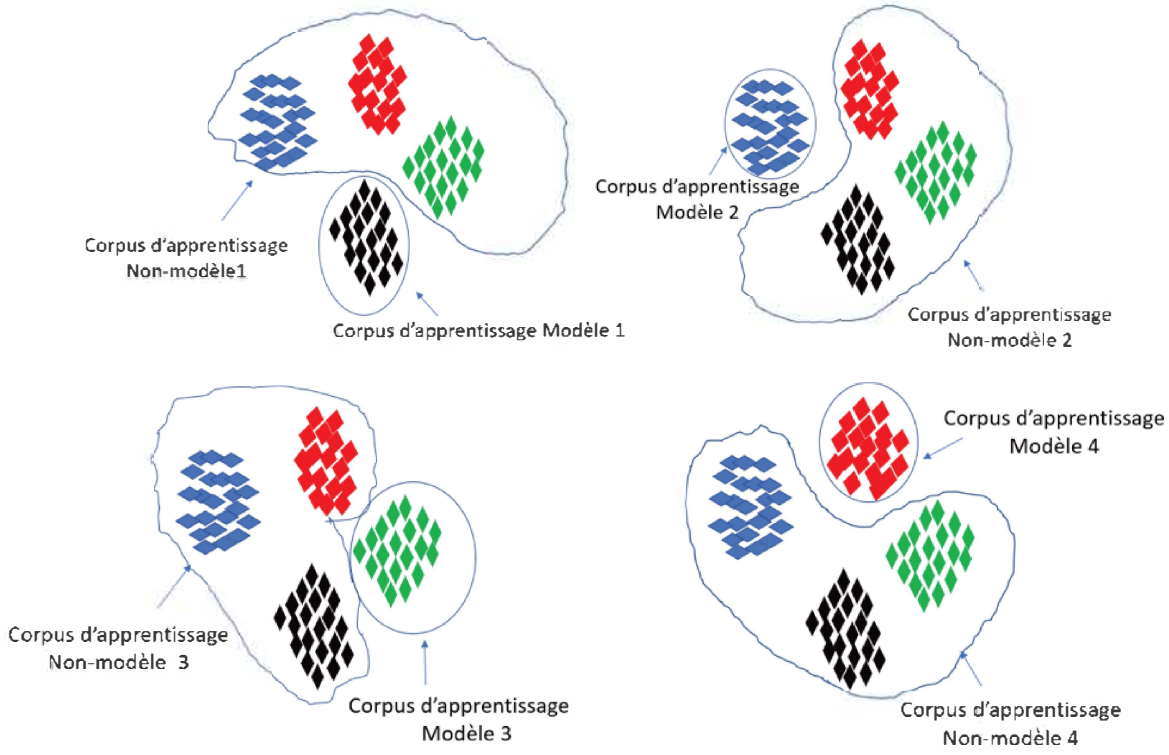

FIGURE 4.2 – Formation des corpus d'apprentissage Non-modèle

4.3.1 Formation des corpus d'apprentissage des Non-modèles

Le processus pour générer les données d'expérimentation a été présenté dans le chapitre 3. Comme nous l'avons expliqué auparavant, nous avons généré plusieurs modèles λ_r avec $r \in [1,R]$. La topologie de chaque modèle est de type gauche-droite à N états avec émission de M symboles différents. La longueur des séquences est T observations. Chaque base de données (O^r) de chaque classe r répartie en base d'apprentissage (O^r_{app}) et base de test (O^r_{test}) dans la proportion de 70% pour l'apprentissage et 30% pour les tests.

Les corpus d'apprentissage pour les Non-modèles ($O_{app\ Nm}$) sont obtenus en concaténant les bases d'apprentissage des (R-1) autres Modèles. Les bases contiennent donc (R-1) fois plus d'éléments que les bases pour apprendre les Modèles. Nous illustrons dans les figures 4.3, 4.4, 4.5 et 4.6 la formation des corpus d'apprentissage des Non-modèles pour un nombre de classe R=4.

$$O^1_{app\ Nm} = \{O^1_{app\ Nm\ 1}, O^1_{app\ Nm\ 2}, \dots, O^1_{app\ Nm\ P}\} \Rightarrow \begin{cases} O^2_{app} = \{O^2_{app\ 1}, O^2_{app\ 2}, \dots, O^2_{app\ L}\} \\ + \\ O^3_{app} = \{O^3_{app\ 1}, O^3_{app\ 2}, \dots, O^3_{app\ L}\} \\ + \\ O^4_{app} = \{O^4_{app\ 1}, O^4_{app\ 2}, \dots, O^4_{app\ L}\} \end{cases}$$

FIGURE 4.3 – Ensemble d'apprentissage du Non-modèle 1

$$(O^2_{app\ Nm}) = \{\{O^2_{app\ Nm\ 1}\}, \{O^2_{app\ Nm\ 2}\}, \dots, \{O^2_{app\ Nm\ P}\}\} \Rightarrow \begin{cases} (\bullet^1_{app}) = \{\{O^1_{app\ 1}\}, \{O^1_{app\ 2}\}, \dots, \{O^1_{app\ L}\}\} \\ + \\ (\bullet^3_{app}) = \{\{O^3_{app\ 1}\}, \{O^3_{app\ 2}\}, \dots, \{O^3_{app\ L}\}\} \\ + \\ (\bullet^4_{app}) = \{\{O^4_{app\ 1}\}, \{O^4_{app\ 2}\}, \dots, \{O^4_{app\ L}\}\} \end{cases}$$

FIGURE 4.4 – Ensemble d'apprentissage du Non-modèle 2

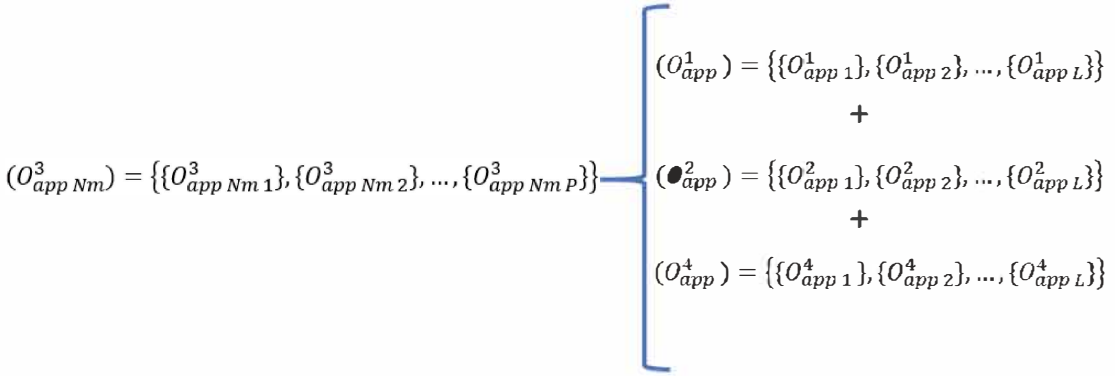

FIGURE 4.5 – Ensemble d'apprentissage du Non-modèle 3

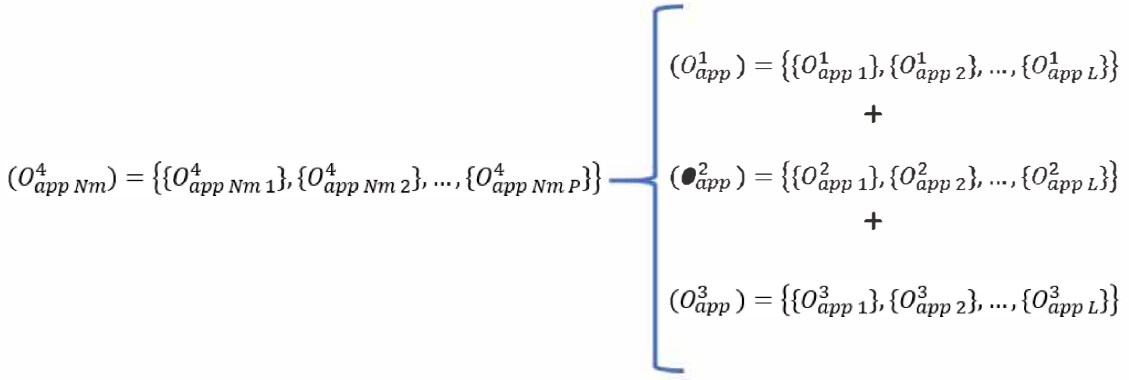

FIGURE 4.6 – Ensemble d'apprentissage du Non-modèle 4

4.3.2 Apprentissage des MMC par les Non-modèles

Une fois les corpus d'apprentissage des Non-modèles créés, nous effectuons l'apprentissage comme nous l'avons effectué dans la méthode de classification usuelle.

Ainsi, l'approche que nous proposons ici consiste à modéliser le système en effectuant l'apprentissage de chacune de ces non classes représentant les Non-modèles $\bar{\lambda}_{Nm}$ par un MMC de même type que pour l'approche usuelle.

La phase d'apprentissage utilise les (R-1) corpus d'apprentissage des Non-modèles pour ajuster les paramètres de chaque MMC de façon à maximiser la probabilité de générer la séquence d'ob-

servations $\{O_{app\ Nm}\} = \{O_{app\ Nm\ 1}, O_{app\ Nm\ 2}, ..., O_{app\ Nm\ T}\}$ étant donné un modèle initial λ. Comme pour l'approche usuelle (chapitre 3), la réponse au problème d'apprentissage est fournie par les formules de réestimation de Baum–Welch ou apprentissage par Viterbi qui, à partir d'un modèle initial et un processus itératif, permettent de maximiser la probabilité $P((O_{app\ Nm})|\lambda)$ afin d'estimer le Non-modèle le plus représentatif des séquences d'observations $(O_{app\ Nm})$.

4.3.3 Phase de Test

Etant donné une séquence à identifier $\{O_{test}\}$ et un Modèle de Markov, nous avons besoin de calculer les probabilités $P((O_{test})|\bar{\lambda}_{Nm})$ pour chaque Non-modèle, afin de déterminer le Non-modèle le plus représentatif (classe d'origine) de cette séquence. Pour la classification, nous avons calculé la vraisemblance pour chaque Non-modèle. Le Non-modèle identifié est celui qui présente la vraisemblance la plus faible : le modèle qui ressemble le moins (ce que l'on est pas).
Les différentes étapes de cette classifcation sont illustrées dans la figure 4.7.

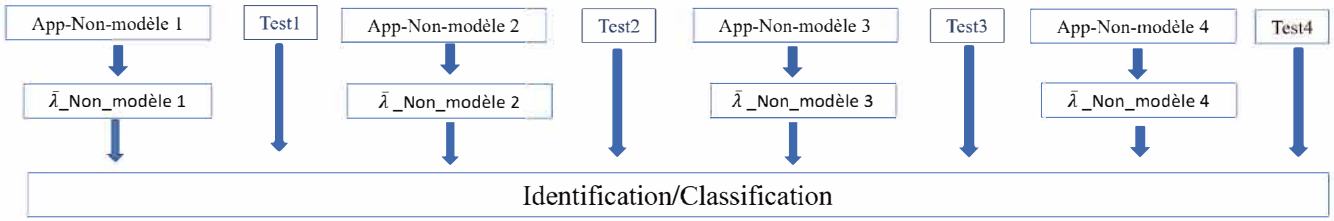

FIGURE 4.7 – Méthode d'identification fondée sur les non-modèles

4.4 Résultats de classification

Nous présentons dans cette section les résultats obtenus pour la classification par Modèles de Markov Cachés sur une base de données test (O^r_{test}).

- Nous avons utilisé les mêmes quatre Modèles λ_r avec r ∈[1,R] ici R=4 générés pour la classification usuelle expliquée dans le chapitre 3.
- La topologie de chaque Modèle est de type gauche-droite à quatre états avec émission de 16 symboles différents.
- La longueur des séquences T = 20 observations.
- Chaque base de données (O^r) contient L_{tot} = 1000 séquences d'observations.

— Les corpus d'apprentissage (O^r_{app}) contiennent L =700 de longueur 20.

— Les corpus de test contiennent (O^r_{test}) $L_{tot} - L$ = 300 séquences d'observations.

— Les corpus apprentissage des Non-modèles ($O^r_{app\ Nm}$) contiennent P = (4-1)*700 =2100 séquences.

Dans un premier temps, nous avons effectué l'apprentissage à l'aide de l'algorithme Baum-Welch. Ensuite, nous avons cherché la vraisemblance de chaque séquence avec chacun des Non-modèles qui ont été estimés $P(\{O^r_{test}\}_l|\bar{\lambda}^r_{Nm})$. Enfin, nous avons effectué l'identification en cherchant le minimum des log-vraisemblances puisque nous avons appris le contraire des séquences (ce que l'on n'est pas). Pour éviter de juger une situation qui pourrait être anecdotique, liée au tirage aléatoire des paramètres des modèles, les tests ont été répétés plusieurs fois. Les résultats obtenus par l'algorithme de Baum-Welch sont présentés dans la figure 4.8.

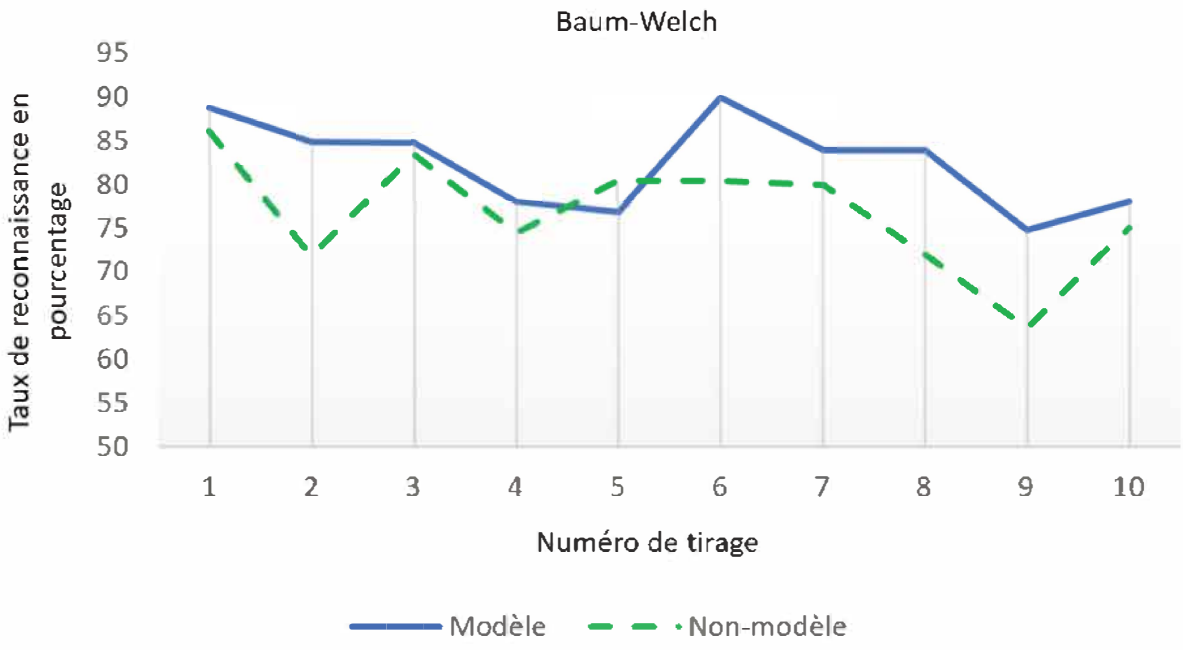

FIGURE 4.8 – Résultat de classification Modèle et Non-modèle avec Baum-Welch

D'après cette figure, nous remarquons que les résultats obtenus par les Non-modèles sont proches de ceux obtenus par les Modèles.

Nous illustrons dans le tableau 4.1 la matrice de confusion de la classification des Non-classes en utilisant l'algorithme de Baum-Welch comme algorithme d'apprentissage.

	Modèles				Non-modèles			
	classe 1	classe 2	classe 3	classe 4	Non-classe 1	Non-classe 2	Non-classe 3	Non-classe 4
Test 1	**95**	3.33	0.33	1.33	**85**	7.33	4	3.66
Test 2	14.66	**72.66**	8.66	4	7.33	**76**	10.33	6.33
Test 3	1.33	1.33	**93.66**	3.66	1.33	2.33	**94**	2.33
Test 4	2.33	2	2.33	**93.33**	4	1.33	5.66	**89**

TABLE 4.1 – Matrice de confusion du premier tirage avec apprentissage Baum Welch

La même démarche a été faite en remplaçant l'algorithme d'apprentissage de Baum-Welch par l'algorithme d'apprentissage par Viterbi. Le résultat obtenu est fourni dans la figure 4.9.

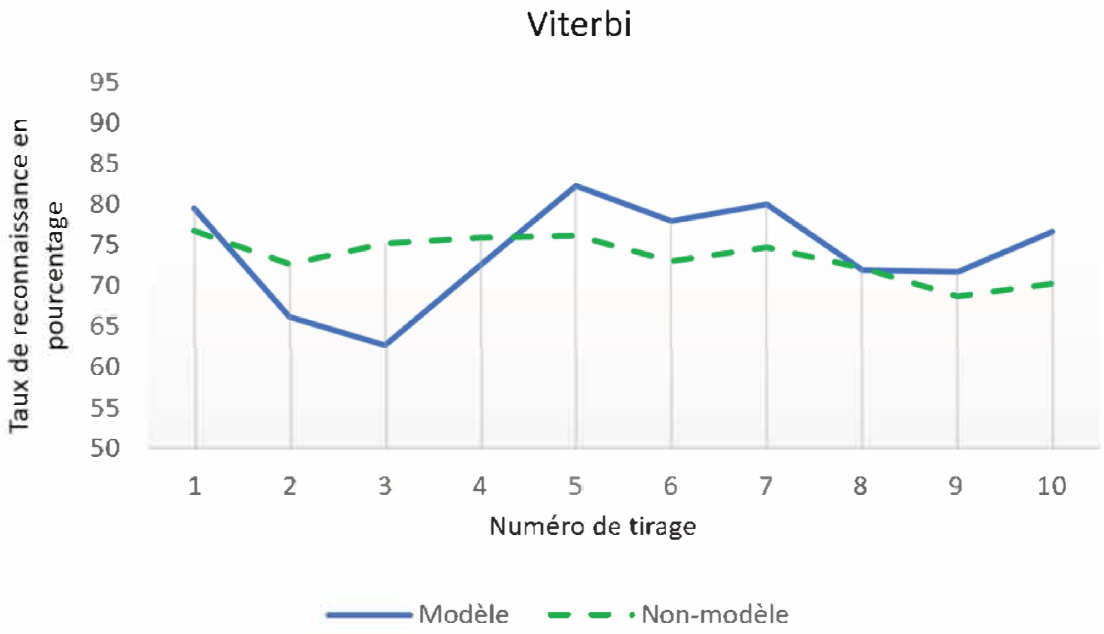

FIGURE 4.9 – Résultats de classification des Modèles et des Non-modèles avec Viterbi

Comme pour l'apprentissage par Baum-Welch, les résultats sont proches. Les résultats fournis par l'apprentissage Viterbi sur les Non-modèles sont un peu inferieurs à ceux obtenus par les Modèles. Le tableau 4.2 montre les taux de classification des quatre classes pour le premier tirage.

	Modèles				Non-modèles			
	classe 1	classe 2	classe 3	classe 4	Non-classe 1	Non-classe 2	Non-classe 3	Non-classe 4
Test 1	**91**	6.33	1.33	1.33	**82.66**	1.66	12	3.66
Test 2	13.66	**72.33**	10.33	3.66	11.33	**59.33**	16	13.33
Test 3	2.33	2.66	**91.66**	3.33	3.66	4	**76**	16.33
Test 4	11	10.66	15.66	**62.66**	1.33	1.33	8.33	**89**

TABLE 4.2 – Matrice de confusion du premier tirage avec apprentissage Baum Welch

4.5 Confiance dans la réponse de classification

Comme nous l'avons fait pour les Modèles, dans cette partie, nous souhaitons visualiser les erreurs et les écarts entre les Non-classes, pour cela, nous avons tracé les vraisemblances normalisées comme nous l'avons fait pour les Modèles.

La différence se fait ici au niveau de l'interprétation des valeurs. Les valeurs de vraisemblances normalisées inférieures à zéro indiquent une erreur. Les valeurs au dessus de zéro montrent l'écart ou la confiance apportée par la bonne réponse.

Les figures 4.10, 4.11 4.12 ainsi que 4.13 montrent ces résultats sur 150 séquences de test, intervalle choisi est limité de -10 à 2.

Sur la figure 4.10, nous remarquons que la majorité des séquences à identifier ont été correctement identifiées (se trouve au niveau 0) ce qui correspond à un taux de reconnaissance global de 82.66% (voir tableau 4.2), les autres séquences mal identifiées se trouvent au-dessous de 0 (car on apprend le contraire de la forme). Nous remarquons aussi que la plupart des erreurs font partie de la Non-classe 3.

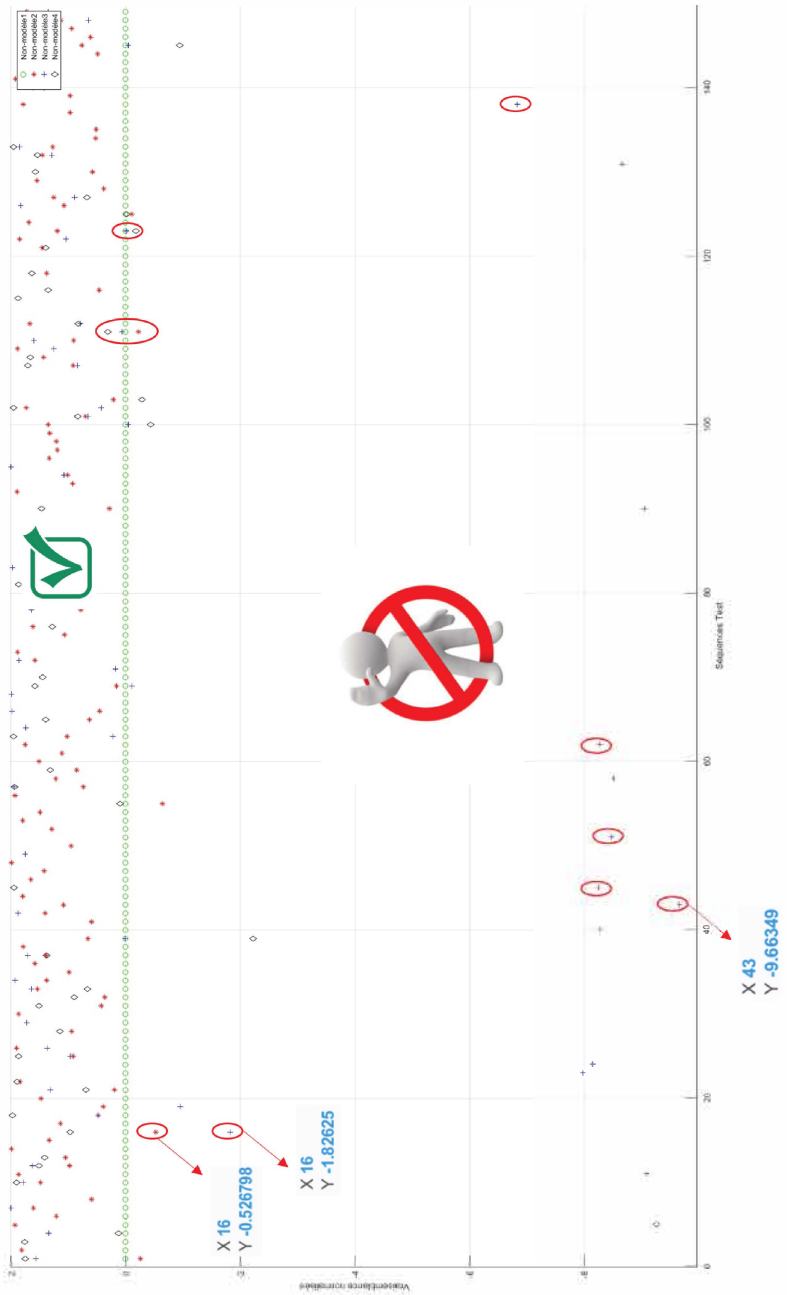

FIGURE 4.10 – Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 1

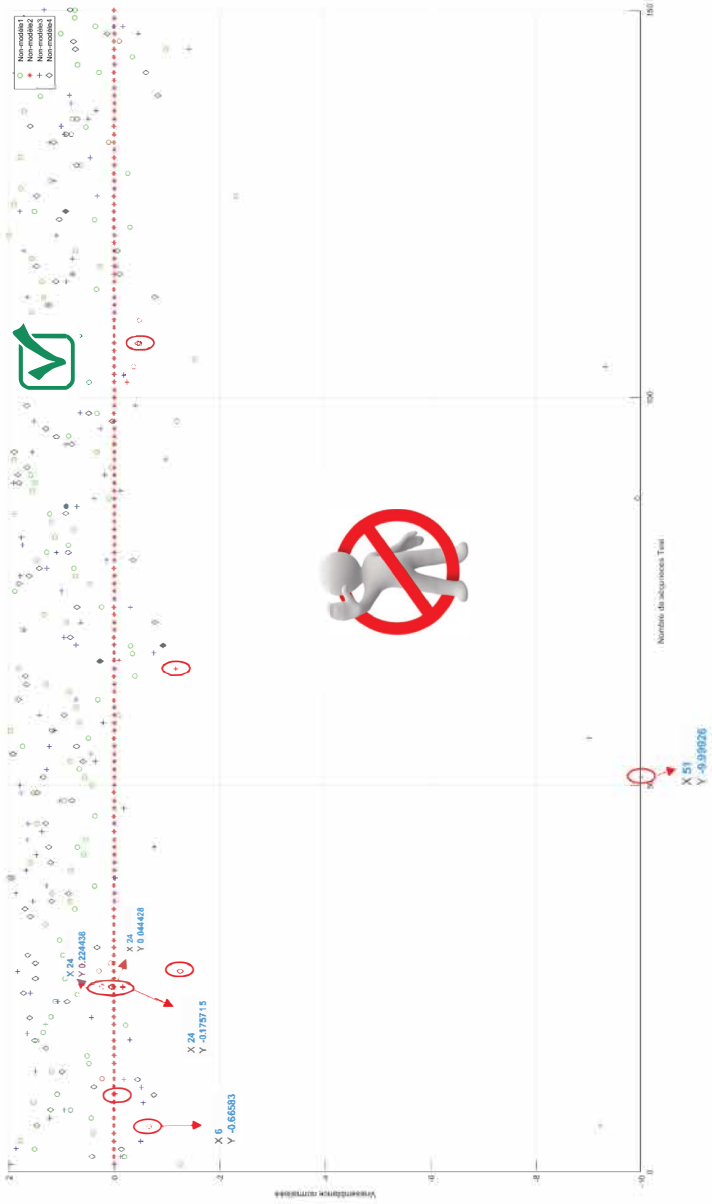

FIGURE 4.11 – Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 2

Sur la figure 4.11, nous remarquons que les écarts sont plus importants par rapport à la figure 4.10, ce qui reflète à un taux de reconnaissance global faible d'environ 59.33% (voir le tableau 4.2).

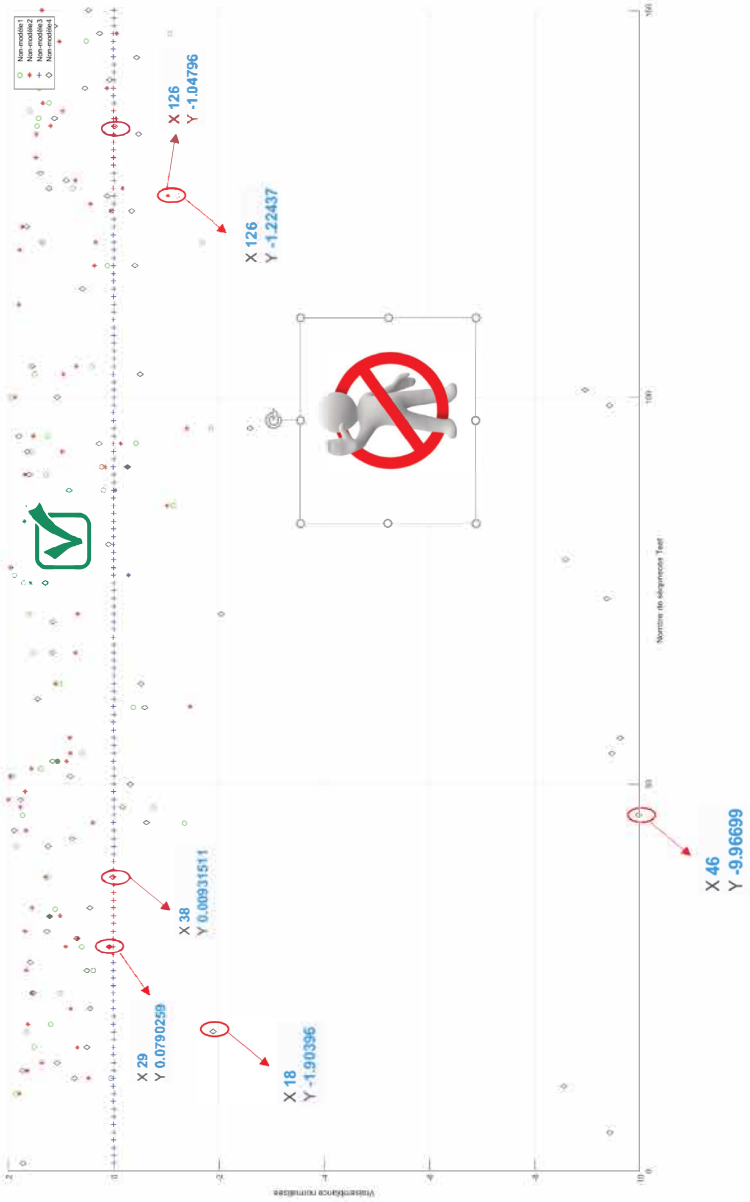

FIGURE 4.12 – Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 3

La figure 4.12 représente la distribution des vraisemblances normalisées des 150 séquences tests pour le Non-modèle 3. Nous remarquons que les écarts sont importants.

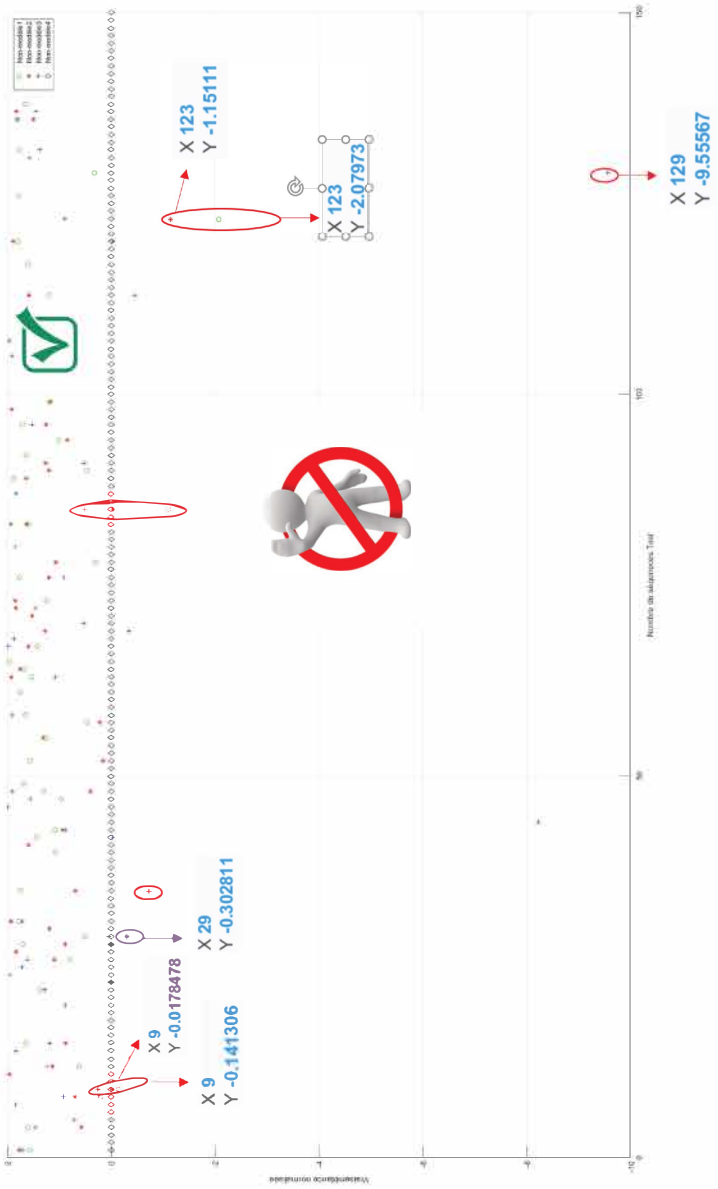

FIGURE 4.13 – Distribution des log-vraisemblances normalisées des 150 séquences tests pour le Non-Modèle 4

Enfin, la La figure 4.13 represente la distribution des vraisemblances normalisées des 150 séquences tests pour le Non-modèle 4. Nous remarquons que les écarts sont moins important par rapport aux deux figures (4.11 et 4.12) avec un taux de reconnaissance global de 89%.

Ces figures nous montrent que le comportement des Non-modèles est assez proche de celui des Modèles. Cependant, la répartition des erreurs est différente.

4.5.1 Discussion sur les résultats obtenus

Les résultats présentés sont encourageants. Mais les informations contenues dans ces « Non-classes » sont-elles complémentaires à celles des classes ? Pour répondre à cette question, nous avons étudié la nature des erreurs commises par les Modèles et les Non-modèles.Nous avons fait un zoom sur la figure 3.10 du chapitre 3 qui représente l'écart entre la classe 1 et les autres classes et 4.10, qui represente l'écart entre la Non-classe 1 et les autres Non-classes. Dans le tableau 4.3, nous montrons un aperçu des erreurs (séquences qui font partie de la classe 1 et non-classe 1 mais, mal identifiés par les Modèles ou Non-modèles) avec apprentissage Viterbi.

	Numéros de séquences à identifier																
Classe 1	**1**	**5**	**11**				**62**										
Classe 2				16				65	103			206	227		253	282	291
Classe 3					55	59											
Classe 4										123	145			242			
Non-classe 1						**59**		**65**				**206**	**227**	**242**	**253**	**282**	**291**
Non-classe 2	1																
Non-classe 3				16	55		62		103								
Non-classe 4		5								123	145						

TABLE 4.3 – Extrait des séquences de la base de test identifiées par les classes et les Non-classes avec apprentissage Viterbi

Ce tableau 4.3 montre différentes situations :

1. Bonne réponse du Modèle, mauvaise réponse du Non-modèle :
   - — Séquences 1,5,11 et 62 : identification correcte pour le Modèle et une erreur commise par le Non-modèle.
2. Mauvaise réponse pour les Modèles et les Non-modèles
   - Même erreur.
     - — Séquences 55 et 62 : même erreur commise par les Modèles et Non-modèles. Nous remarquons que les deux séquences sont considérées de la classe 3 et la Non-classe 3.

- Erreurs différentes :
  - Séquences 16 et 103 : les deux séquences sont mal identifiées par les Modèles et Non-modèles, mais l'erreur n'est pas la même. Nous remarquons que les séquences 16 et 103 sont considérées de la classe 2 par les Modèles et elles sont considéréés de la Non-classe 3 par les Non-modèles.

3. Mauvaise réponse du Modèle mais bonne réponse du Non-modèle :
   - Les séquences 59, 65, 206, 227, 242, 253, 282 et 291 : ces séquences sont mal identifiées par les Modèles mais elles sont bien identifiées par les Non-modèles.

A l'aide de ces résultats, nous pouvons constaté la capacité des Non-modèles à identifier correctement des échantillons.
Dans la section suivante, nous allons montrer comment exploiter l'information contenue dans les Non-modèles afin d'améliorer l'apprentissage des Modèles.

4.6 Algorithme d'apprentissage Non-Behavior Viterbi (NB Viterbi)

Beaucoup de travaux ont été effectués dans le but d'améliorer l'apprentissage des MMC, en effet Lember et al dans [74] ont apporté des modifications de l'algorithme d'apprentissage de Viterbi pour estimer les paramètres d'émission des Modèles de Markov Cachés (MMC) en général, et dans les modèles de mixité en particulier. Pour cela, ils ont ajusté l'algorithme d'apprentissage de Viterbi afin d'améliorer considérablement la précision sans pour autant coûter trop cher en temps de calcul. En même temps, ils ont proposé une deuxième version de cette modification qui est une correction mathématiquement plus avancée de Viterbi, qui permet une convergence plus rapide et une plus grande précision.
Dans [75], les chercheurs ont présenté une combinaison d'apprentissage de Viterbi et de Baum-Welch conçue dans le cadre de la reconnaissance de la parole. La combinaison proposée de Viterbi et de Baum-Welch ne surpasse l'apprentissage de Viterbi que dans le cas de la parole lue à 8 kHz.
Les auteurs dans [76] ont proposé, dans le cadre de la détection de symboles, un algorithme qui s'appelle ViterbiNet. C'est un détecteur de symboles piloté par les données, obtenu en convertissant l'algorithme de Viterbi en un système utilisant des réseaux neuronaux profonds (DNN). Contrairement à l'algorithme de Viterbi, ViterbiNet est robuste à l'incertitude du modèle.

Z. Baklouti dans [77] a développé une recherche dans le but d'identifier la partie des phrases d'un tweet qui suscite n'importe quelle émotion. Pour atteindre cet objectif, il a amélioré l'algorithme de Viterbi. Cette recherche est parvenu à un developpement d'un modèle plus facilement compréhensible et en incorporant des avancées récentes grâce à des techniques d'apprentissage par transfert pour le rendre capable de recevoir un modèle pré-entraîné.
Sassi et al dans [78] ont proposé un nouvel algorithme de Viterbi parallèle rapide et contraint pour les données volumineuses. Ils ont fourni une analyse détaillée de ses performances dans le cadre de big data. Cette analyse des performances comprend l'évaluation du temps d'exécution, de la vitesse et de la précision de la prédiction.

Dans notre étude, l'objectif est d'améliorer l'apprentissage des Modèles de Markov Cachés et après avoir montré que les Non-modèles ont la capacité d'identifier correctement les séquences inconnues, nous souhaitons exploiter l'information contenue dans ces Non-modèles afin d'améliorer l'apprentissage des Modèles.

Pour répondre à cette problématique, nous avons développé un nouvel algorithme qui s'appuie sur le principe de l'algorithme de Viterbi. Cet algorithme sera détaillé dans la section suivante.

4.6.1 Explication de l'algorithme NB Viterbi

Dans l'algorithme d'apprentissage usuel de Viterbi, l'idée générale de la procédure d'apprentissage est simple : l'algorithme de Viterbi est utilisé pour identifier chaque séquence d'état selon le modèle courant, les meilleurs chemins de Viterbi sont utilisés pour améliorer les probabilités en incrémentant chaque transition comme nous l'avons montré dans le chapitre 2. Plus une séquence d'états est fréquente, plus la probabilité associée est élevée.
Le principe que nous voulons mettre en oeuvre dans ce nouvel algorithme est de prendre en compte à la fois l'information contenue dans les Modèles et celle des Non-modèles.
Afin de transposer l'effet des séquences d'états de la base 'Modèle' qui consiste à incrémenter les compteurs, les séquences d'états pour les 'Non-modèles' vont décrémenter les compteurs.
Dans les données Non-modèles, plus une séquence d'états est fréquente, plus la probabilité associée est faible. Nous illustrons le chemin de Viterbi dans la figure 4.14.

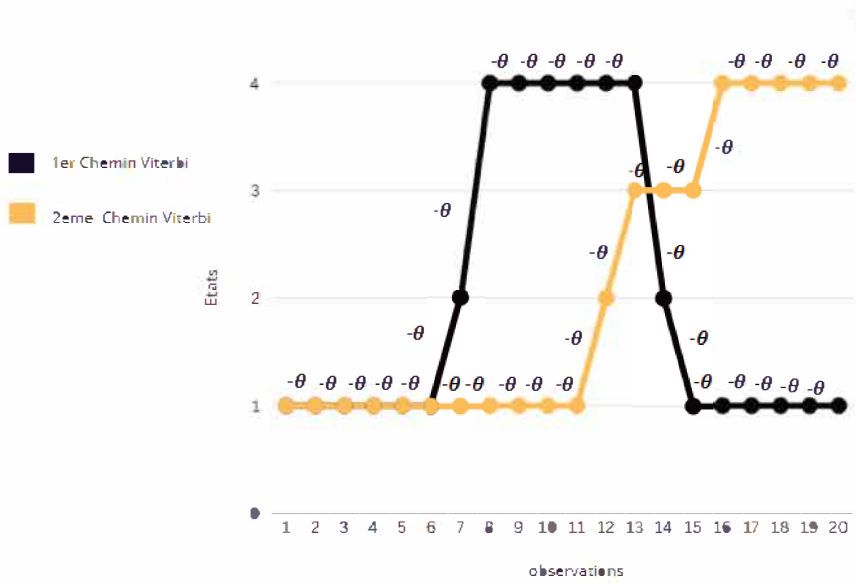

FIGURE 4.14 – Illustration des meilleurs chemins de Viterbi pour les Non-modèles pour deux séquences

Chaque fois que nous passons d'un état à un autre, nous incrémentons les variables n_i, $n_{i\to j}$ et n_o, ces variables représentent respectivement le nombre de fois ou l'état i est utilisée, le nombre de fois où la transition $i \to j$ est utilisé, et enfin le nombre de fois ou le symbole V_k est généré par l'état i. Pour les Modèles, nous incrémentons de $(+1)$ et pour les Non-modèles nous décrementons de (θ). Le choix de la valeur de (θ) permet de régler le poids attribué aux Non-modèles par rapport aux Modèles. La cellule de $n_{i\to j}$ du chemin de Viterbi est indiquée dans le tableau 4.4

Vers l'état j / *De l'état i*	1	2	3	4
1	-5 θ -10 θ	- θ- θ	0	0
2	$-\theta$	0	- θ	$-\theta$
3	1	0	-2 θ	-θ
4	0	-θ	0	-5θ-4θ

TABLE 4.4 – Contribution des deux chemins de Viterbi

Finalement, nous obtenons la somme des deux contributions et comme nous incrémentons d'une part et nous décrémentons d'autre part, nous risquons d'avoir des valeurs négatives. Pour éviter

ce problème, nous avons rajouté un offset aux variables n_i, $n_{i\to j}$ et n_o comme suit :

$$Offset_{n_i^o} = abs(min(n_i^o)) \tag{4.1}$$

$$Offset_{n_{i\to j}} = abs(min(n_{i\to j})) \tag{4.2}$$

Enfin, nous réestimons les paramètres jusqu'à la stablisation. L'algorithme NB Viterbi est illustré dans l'algorithme 6.

Algorithme 6 : Apprentissage par NB Viterbi

Données : $(O_{app}) = \{\{O_{app\ 1}\},\{O_{app\ 2}\},...,\{O_{app\ L}\}\}$,
$(O_{app\ Nm}) = \{\{O_{app\ Nm\ 1}\},\{O_{app\ Nm\ 2}\},...,\{O_{app\ Nm\ P}\}\}$

Choisir un modèle initial $\lambda(A,B,\pi)$
Initialiser les variables : $n_i, n_{i\to j}$ *et* n_i^O
pour *chaque* $\{O_{app\ l}\}$ *dans* (O_{app}) **faire**
    Calculer le chemin Viterbi de $\{O_{app\ l}\}$ Ajuster les variables $n_i, n_{i\to j}$ *et* n_i^O avec la contribution de $\{O_{app\ l}\}$
fin
pour *chaque* $\{O_{app\ Nm\ l}\}$ *dans* $(O_{app\ Nm})$ **faire**
    Calculer le chemin Viterbi de $\{O_{app\ Nm\ l}\}$ Ajuster les variables $n_i, n_{i\to j}$ *et* n_i^O avec la contribution de $\{O_{app\ Nm\ l}\}$
fin
Ajouter l'offset aux valeurs $n_i, n_{i\to j}$ *et* n_i^O afin d'éviter les éléments négatifs
Réestimer les paramètres $\lambda(A,B,\pi)$ en utilisant les formules 2.32, 2.33
Jusqu'au critère de fin

4.7 Résultats obtenus par NB Viterbi

Nous présentons dans cette section les résultats obtenus pour la classification en utilisant l'algorithme d'apprentissage NB Viterbi. Ici, nous utilisons les deux corpus d'apprentissage (O_{app}) et ($O_{app\ Nm}$). Les tests ont été effectués sur les mêmes corpus test (O^r_{test}). Comme pour la méthode de classification usuelle, nous avons répété les tests une dizaine de fois. La figure 4.15 montre les résultats de classification pour les deux algorithmes d'apprentissage (Viterbi et NB Viterbi).

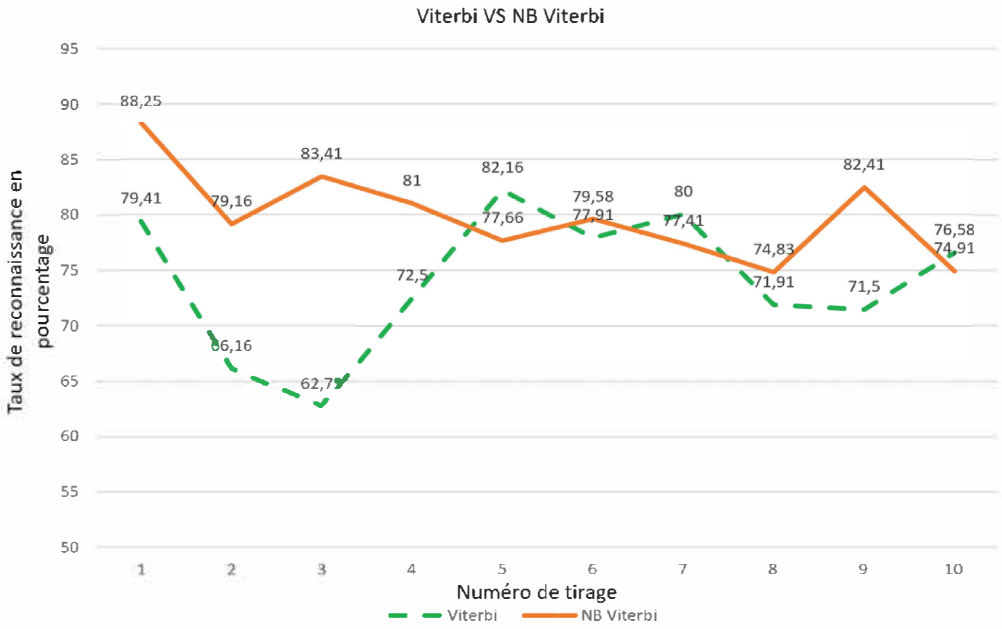

FIGURE 4.15 – Résultat de classification sur dix tirages aléatoires

D'après la figure 4.15, nous remarquons que le nouvel algorithme NB Viterbi offre globalement des meilleurs résultats que ceux obtenus par l'algorithme d'apprentissage de Viterbi. Nous remarquons aussi sur le deuxième tirage que le taux de reconnaissance obtenue par l'apprentissage NB Viterbi dépasse le résultat obtenu par apprentissage de Viterbi de 13%.

Nous illustrons dans le tableau 4.5 la matrice de confusion obtenue après apprentissage par l'algorithme de NB Viterbi, pour le premier jeu de données (premier tirage). Le pourcentage global obtenu est de 88.25% (la moyenne de la diagonale en rouge). Ce chiffre correspond au premier tirage de la figure 4.15.

		Classe 1	Classe 2	Classe 3	Classe 4
Classe1	NB Viterbi	**89.66**	3	4.33	3
	Viterbi	**91**	6.33	1.33	1.33
Classe 2	NB Viterbi	10	**73.66**	9	7.33
	Vitrbi	13.66	**72.33**	10.33	3.66
Classe 3	NB Viterbi	0.66	1	**96.66**	1.66
	Vitrbi	2.33	2.66	**91.66**	3.33
Classe 4	NB Viterbi	2.66	1.33	3	**93**
	Vitrbi	11	10.66	15.66	**62.66**

TABLE 4.5 – Matrice de confusion du premier tirage avec apprentissage NB Viterbi

4.7.1 Comparaison des résultats obtenus par apprentissage NB Viterbi avec les autres algorithmes d'apprentissage

Dans cette partie, nous allons comparer les résultats obtenus par l'apprentissage NB Viterbi et les autres algorithmes d'apprentissage (Viterbi et Baum-Welch). Pour cela, nous avons tracé trois courbes qui representent les résultats obtenus par les algorithmes d'apprentissage sur les dix tirages (figure 4.16).

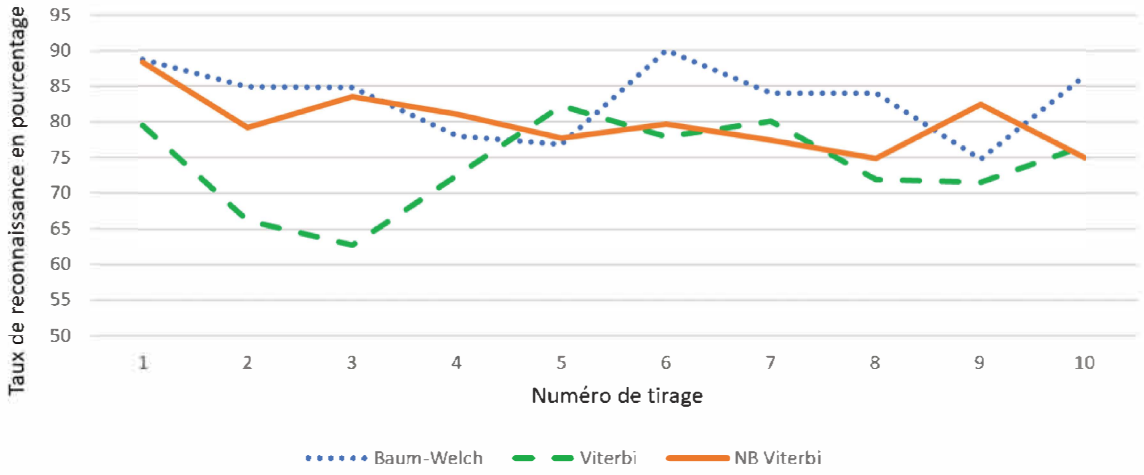

FIGURE 4.16 – Résultats obtenus par les trois algorithmes d'apprentissage sur dix tirage

D'après cette figure, nous remarquons que les résultats obtenus par NB Viterbi sont globalement meilleurs que ceux obtenus par l'algorithme d'apprentissage Viterbi. Nous remarquons aussi que les résultats sur certains tirages sont identiques aux résultats obtenus par l'algorithme d'apprentissage de Baum-Welch (exemple : tirage 1) et sur certains tirages les résultats obtenus par NB Viterbi sont meilleurs que ceux obtenus par apprentissage de Baum-Welch (tirages 3, 4 et 9).

4.7.2 impact des corpus Non-Modèle sur les résultats obtenus

Le nouvel algorithme développé (NB Viterbi) exploite les informations contenues dans les Modèles ainsi que les informations contenues dans les Non-modèles. Afin de vérifier l'influence des Non-modèles sur les Modèles, nous avons fait varier le poids des Non-modèles de 0% à 300%. La définition du poids est comme suit :

— Pour un poids de 0% : l'algorithme n'utilise que les informations contenues dans les Modèles.

— Si le poids des Non-modèles est à 100% : l'algorithme utilise autant d'information contenue dans les Non-modèles que d'information contenue dans les Modèles.

— Si le poids des Non-modèles est à 300% : cela veut dire que l'on donne 3 fois plus d'importance aux Non-modèles.

La figure 4.17 montre les résultats obtenues en variant le poids des Non-modèles de 0% à 300%. Nous remarquons que le meilleur taux de reconnaissance est obtenu lorsque les Non-modèles sont entre 75% et 100%.

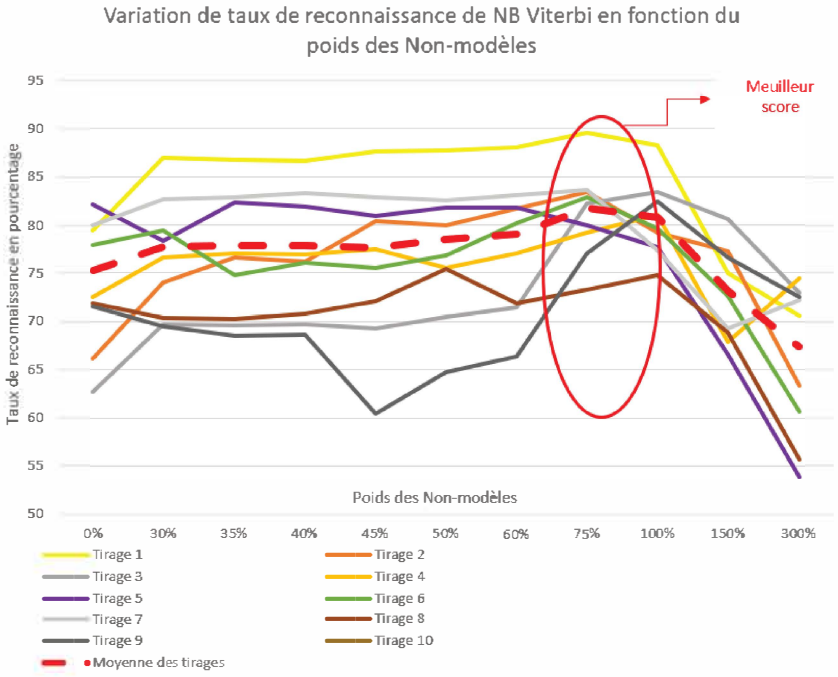

FIGURE 4.17 – Variation du poids des Non-modèles

4.8 Variation du nombre d'états des MMC

Afin de vérifier la robustesse de notre algorithme NB Viterbi, nous avons fait varier le nombre d'états N de 4 à 10 états par Modèle. Nous avons choisi les poids 75% et 100% pour les Non-modèles. Les résultats obtenus sont donnés dans la figure 4.18.

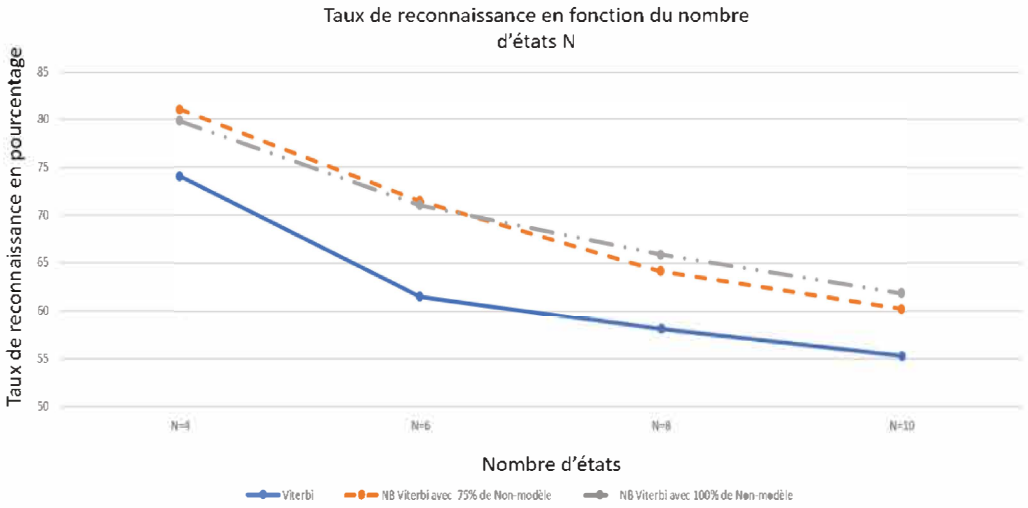

FIGURE 4.18 – Variation du nombre d'états avec apprentissage Viterbi et NB Viterbi

Sur cette figure 4.18, nous remarquons que plus on augmente le nombre d'états plus le taux de

reconnaissance diminue. Cela peut être dû à la compléxité du système qui augmente.
Cependant, nous remarquons que malgré l'augmentation du nombre d'états, l'algorithme NB Viterbi reste meilleur que l'algorithme d'apprentissage de Viterbi avec un dépassement dans certain cas de plus de 9.93% (figure 4.19). Cela signifie que les Non-modèles apportent une amélioration dans tous les cas testés.

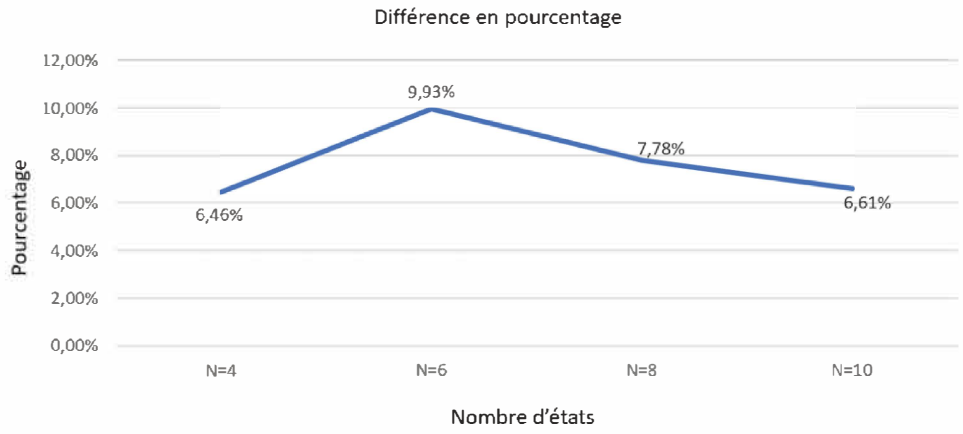

FIGURE 4.19 – différence des taux de reconnaissance en pourcentage entre Viterbi et NB Viterbi

4.9 Conclusion

Dans ce chapitre, nous avons montré une nouvelle méthode d'apprentissage des Modèles de Markov Cachés. Nous avons fait un état de l'art sur les différentes méthodes d'apprentissage des MMC. Nous avons introduit la stratégie d'apprentissage qui consiste à apprendre le contraire d'une forme. Nous avons d'abord montré comment les corpus d'apprentissage sont construits. En effet, chaque corpus d'apprentissage Modèle a un pendant qui est la concaténation de tous les échantillons des Non-modèles. Ensuite, nous avons effectué l'apprentissage à l'aide des algorithmes Baum-Welch et apprentissage par Viterbi. Les résultats obtenus montrent que les Non-modèles ont la capacité d'identifier correctement les classes.

Par la suite, nous avons vérifié que les Non-modèles contiennent de l'information pertinente. Nous avons donc effectué cette vérification en étudiant les erreurs commises par la classification des

Non-modèles et nous avons montré que certains echantillons sont mal identifiés par les Modèles mais ils sont correctement identifiés par les Non-modèles.

En s'appuyant sur les résultats obtenus par les Non-modèles, nous avons intégré l'information contenues dans les Non-modèles afin d'améliorer l'apprentissage des Modèles. Pour cela, nous avons développé un nouvel algorithme : le NB Viterbi qui consiste à utiliser à la fois les corpus d'apprentissage des Modèles ainsi que les corpus d'apprentissage des Non-modèles. Les résultats obtenus par cet algorithme sont trés intéressants et montrent que le nouvel algorithme permet parfois d'obtenir des résultats superieurs à ceux de l'algorithme d'apprentissage Baum-Welch.

Ensuite, afin d'etudier l'influence du corpus d'apprentissage Non-modèles, nous avons fait varier le poids de ce dernier entre 0% *et* 300%, et nous avons montré que les meilleurs taux de reconnaissance sont obtenus lorsqu'on utilise entre 75% *et* 100% de Non-modèles.

Conclusion Générale

Les travaux présentés dans cette thèse traitent les problèmes d'apprentissage multi-classes afin d'estimer l'état de santé des systèmes industriels. En effet, un système industriel qui présente plusieurs modes de défaillance auquels un mode sans défaillance s'ajoute peut être vu comme un problème multi-classes. Pour cela, nous employons les méthodes de classification et d'identification afin d'effectuer le diagnostic des pannes. Le diagnostic permet de déterminer en ligne les composants du système qui sont en fautes et donc qui doivent être réparés. Le pronostic permet d'anticiper les défauts en fournissant les informations sur l'état futur du système à partir duquel des actions de maintenance peuvent être envisagées.

Nous avons débuté cette étude, au Chapitre 1, en présentant un état de l'art sur les différentes stratégies de maintenance utilisées pour assurer la fiabilité et la sûreté de fonctionnement des systèmes industriels. Le but de cette étude était de positionner le diagnostic et le pronostic au sein de la maintenance préventive conditionnelle. Ce premier chapitre a permis de mettre en évidence deux grandes catégories d'approches. La première approche utilise un modèle mathématique fondé sur les résidus pour détecter et suivre la dégradation du système. Cette approche est bien adaptée lorsqu'un modèle analytique précis peut être construit. Cependant, dans la réalité, la complexité du système est telle qu'extraire un modèle est souvent difficile à mettre en oeuvre ou dans certains cas impossible à extraire. La deuxième approche consiste à analyser les signaux de mesure pour extraire un vecteur de caractéristiques d'état de santé. Ce vecteur est ensuite utilisé dans la reconnaissance des formes afin de détecter des changements dans les données de mesure à travers des méthodes de classification et d'identification. À des fins de diagnostic et de pronostic, des modes de fonctionnement ont été définis pour représenter le comportement du système. Dans le cas du diagnostic, ces modes de fonctionnement peuvent représenter les diffé-

rents types de défauts qu'un système peut rencontrer durant son fonctionnent.

Le second chapitre a présenté l'approche que nous avons utilisé pour le diagnostic et le pronostic des défaillances. Cette approche dite « markovienne » fait appel au Modèles de Markov Cachés utilisés pour la modélisation des comportements. Nous avons vu à travers la visualisation réseaux avec le logiciel VOSViewer qu'il existe une très forte relation entre la maintenance préventive conditionnelle et les Modèles de Markov Cachés. En effet, les MMC sont des outils statistiques permettant de modéliser de nombreux phénomènes stochastiques. Ils sont utilisés dans de nombreux domaines avec efficacité. Dans ce chapitre, nous avons présenté les différentes architectures ainsi que les différents problèmes des MMC. Nous avons aussi introduit les algorithmes d'apprentissage (Baum-Welch et Viterbi). Ensuite, nous avons effectué une petite comparaison entre les deux principaux algorithmes d'apprentissage des MMC à savoir Baum-Welch et apprentissage par Viterbi. Enfin, nous avons défini une stratégie d'apprentissage afin de résoudre la problématique de classification multi-classes.

Le troisième chapitre de cette thèse consiste à appliquer les méthodes de diagnostic et de pronostic en utilisant la classification par MMC. Pour cela, nous avons généré des modèles de synthèse. A partir de ces modèles, nous avons généré nos bases de données. Ces données représentent des modes de défaillance plus un mode sans défaillance. Ensuite, nous avons séparé ces données en corpus d'apprentissage et de test. Nous avons également étudié l'homogénéité de ces corpus pour chacune des classes.
Nous avons expliqué la classification usuelle en utilisant les Modèles de Markov Cachés. L'apprentissage a été fait à l'aide des algorithmes d'apprentissage de Baum-Welch et Viterbi. Les résultats de cette méthode serviront de référence pour la suite de notre étude.
Enfin, nous avons étudié la pertinence des données en employant les techniques de la théorie de l'information (entropie de Shannon et l'information mutuelle).

Dans la dernière partie de cette thèse (chapitre 4), nous avons répondu aux principales problématiques de cette thèse. Pour cela, nous avons introduit une nouvelle technique d'apprentissage des MMC. Cette dernière consiste à apprendre le contraire d'une forme. Nous avons d'abord formé les nouveaux corpus d'apprentissage Non-modèles. Puis, nous avons effectué l'apprentissage avec

les algorithmes de Baum-Welch et Viterbi. Enfin, nous avons montré que ces Non-modèles ont la capacité à discriminer correctement les classes. Cette méthode a permis de répondre à la première problématique.

Nous avons ensuite exploité les deux corpus d'apprentissage Modèles et Non-modèles afin d'améliorer l'apprentissage des MMC. Nous avons développé un nouvel algorithme : NB Viterbi qui consiste à exploiter l'information contenu dans les Modèles et dans les Non-modèles. Nous avons montré que cet algorithme apporte de meilleurs résultats que l'algorithme de Viterbi. Nous avons montré aussi que les meilleurs résultats sont obtenus en pondérant les Non-modèles à hauteur de 75%.

Enfin, nous avons montré qu'en augmentant la compléxité du système, l'algorithme NB Viterbi améliore les résultats.

Situation scientifique dans le contexte National et International actuel

Il existe différentes approches récentes dans la maintenance industrielle afin de résoudre la problématique d'anticipation des pannes par diverses techniques comme :

- Choisir de bonnes stratégies de fiabilité i.e. établir une politique de surveillance sur les équipements les plus sensibles aux risques de défaillances, en intégrant par exemple des systèmes de télésurveillance industrielle dans un système intégré connecté à une plateforme logicielle fondée sur le cloud, cela permet une visibilité critique, recevoir des alertes indiquant des anomalies ;
- planifier le choix des tâches de maintenance en fonction des conséquences des défaillances ;

Dans notre démarche, nous tentons d'effectuer le pronostic et le diagnostic des pannes en présentant des outils de choix et d'optimisation de modélisations markoviennes en considerant les différentes défaillances comme un problème multi-classes.

Bénéfices et originalité de notre approche fondée sur l'approche Markovienne :

Les méthodes de classification pour le diagnostic et le pronostic des défauts reposent sur le classement d'observations (objets ou formes) en les comparant à des formes-types. Notre approche est fondée sur l'utilisation des MMC pour la détection de défaillances. Du point de vue de l'expert en maintenance, l'intérêt fondamental de notre approche est son aspect visuel qui permettra de contrôler facilement l'état de son système. En outre, bénéficiant de toute la théorie Markovienne, cette approche reste très efficace pour l'estimation des défaillances et assez simple à mettre en œuvre dans un système de gestion de la maintenance industrielle informatisée. Son originalité réside sur les points suivants :

— Nous avons montré qu'il était possible d'estimer le niveau de dégradation d'un système à partir d'informations observables en utilisant une modélisation à partir d'un Modèle de Markov Caché.

— Nous avons montré que le cadre que notre recherche se positionne sur des problématiques de type multiclasses.

— Le Non-comportement : nous avons montré que les non-comportements appelés dans cette thèse "Non-modèles" contiennent l'information nécessaire afin d'estimer correctement les classes.

— Le contenu de ces non-comportements est complémentaire aux informations contenues dans le comportement. D'ailleurs, nous avons développé un algorithme qui permet d'exploiter les deux informations pour l'apprentissage des MMC.

Développement durable

Dans nos travaux, nous determinons une bonne méthode de classification et d'identification afin de diagnostiquer et prédire les pannes industrielles. Cette methode devrait permettre d'abaisser significativement le taux de pannes du matériel de production. Ce qui veut dire moins d'intervention humaine et donc moins de risques. En effet, les enjeux du développement durable sont liés à la notion de risque, qu'il s'agisse de risques pour les humains, de risques pour l'environnement ou de risques financiers. Pour atteindre les objectifs du développement durable, il est donc naturel de s'appuyer sur une analyse des risques et une maîtrise des actions mises en place et de leurs conséquences sur ces risques. Or, les activités de la fonction maintenance sont au cœur de la maîtrise de ces risques puisque l'objectif de notre méthode est de maîtriser le système technique et de prendre les meilleures décisions pour l'entreprise afin d'éviter l'occurrence d'événements

initiateurs dans la chaîne de propagation des risques. Les accidents industriels passés rappellent à quel point la fonction maintenance a un rôle primordial lors de l'exploitation des systèmes. La fonction maintenance est également actrice des décisions en fin d'exploitation ou en fin de vie des systèmes, elle est donc en relation directe avec la nécessité de recycler et revaloriser les équipements obsolètes.

Les limites de l'approche

Lors de l'utilisation des MMC, un certain nombre de problèmes de calculs numériques surviennent. Ils sont plus ou moins associés à la taille du corpus d'apprentissage, à la discrétisation des symboles utilisés. On peut retenir :

— l'application de notre méthode sur des données industrielles réelles. En effet, dans notre thèse, nous avons utilisé des données de synthèse. Malheureusement c'est difficile de trouver des bases de données libres en accès avec plusieurs modes de defaillances.

— l'insuffisance de données pour l'apprentissage,

— l'initialisation du MMC durant la procédure d'apprentissage.

Concernant l'insuffisance de données pour l'apprentissage : pour que les paramètres du MMC convergent vers les vraies valeurs, on doit avoir un nombre d'observations qui tend vers l'infini [38]. En théorie, la solution est d'augmenter la taille du corpus d'apprentissage.
Concernant l'initialisation et la détermination du nombre d'états d'un MMC, c'est un choix très difficile. Il n'existe pas de méthode théorique permettant de déterminer le nombre exact d'états pour l'optimisation du MMC. De plus, les états ne sont pas nécessairement liés à un ou des phénomènes physiques observables. Pour le moment, il n'existe aucune méthode qui determine le nombre de symboles en fonction du nombre d'états. Rabiner [40] propose l'utilisation des connaissances a priori du domaine pour la détermination de l'architecture du MMC.

Les perspectives d'approfondissement

Les travaux réalisés dans le cadre de cette thèse pourraient donner lieu à des études supplémentaires dont nous présentons quelques points qui nous semblent important. Ils sont résumés comme suit :

— Application de notre nouvel algorithme d'apprentissage NB Viterbi sur des bases de données

réelles,

— Tests avec des variations de paramètre (nombre de symboles, la longueur des séquences,...),

— Utilisation de notre nouvelle stratégie d'apprentissage en utilisant le principe de l'algorithme de Baum-Welch.

Bibliographie

[1] L.S. Pluie de produits chimiques, des résidus verts à "désactiver" : inquiétude près de chicago après un énorme incendie dans une usine. *LA DEPECHE*, 12 janvier 2023.

[2] Bernard Jaguenaud. La gestion des risques industriels passée au crible de la cour des comptes. *Face au risque, le média de référence des résponsables de la sécurité*, 2024.

[3] Mathilde Lebrun. Maintenance industrielle en France : répondre aux nouveaux en jeux. 2023.

[4] Normes nationales et documents normatifs nationaux. page 99 p, janvier 2018.

[5] R. Isermann. Supervision, fault-detection and fault-diagnosis methods — an introduction. *Control Engineering Practice*, 5(5) :639–652, 1997.

[6] Rommert Dekker. Applications of maintenance optimization models : a review and analysis. *Reliability Engineering & System Safety*, 51(3) :229–240, 1996. Maintenance and reliability.

[7] Wallace R Blischke, DN Prabhakar Murthy, et al. *Case studies in reliability and maintenance*, volume 480. Wiley Online Library, 2003.

[8] Hongzhou Wang. A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139 :469–489, 02 2002.

[9] Thyago P. Carvalho, Fabrízzio A. A. M. N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto, and Symone G. S. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137 :106024, 2019.

[10] Kamran Javed, Rafael Gouriveau, and Noureddine Zerhouni. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mechanical Systems and Signal Processing*, 94 :214–236, 09 2017.

[11] C. Hohmann. *Téchniques de de productivité*. Livres outils performance, 2009.

[12] Andrew K.S. Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7) :1483–1510, 2006.

[13] John F. Macgregor James V. Kresta and Thomas E. Marlin. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69 :35–47, 1991.

[14] Barry M. Wise and Neal B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6) :329–348, 1996.

[15] Vincentius Ewald, Ramanan Sridaran Venkat, Aadhik Asokkumar, Rinze Benedictus, Christian Boller, and Roger M Groves. Perception modelling by invariant representation of deep learning for automated structural diagnostic in aircraft maintenance : A study case using deepshm. *Mechanical Systems and Signal Processing*, 165 :108153, 2022.

[16] Yusung Lee and Woohyun Kim. Fault detection and diagnosis for variable refrigerant flow systems by using virtual sensors and deep learning. *Energy Reports*, 11 :471–482, 2024.

[17] Olivier Ondel, Emmanuel Boutleux, and Clerc Guy. A method to detect broken bars in induction machine using pattern recognition techniques. *Industry Applications, IEEE Transactions on*, 42 :916 – 923, 08 2006.

[18] R. Casimir, Emmanuel Boutleux, Guy Clerc, and F. Chappuis. Broken bars detection in an induction motor by pattern recognition. *2003 IEEE Bologna Power Tech Conference Proceedings,*, 2 :7 pp. Vol.2–, 2003.

[19] G. Schreier, J. Ragot, R.J. Patton, and P.M. Frank. Observer design for a class of non-linear systems. *IFAC Proceedings Volumes*, 30(18) :483–488, 1997. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 97), Kingston upon Hull, UK, 26-28 August 1997.

[20] Rolf Isermann. *Fault-diagnosis systems : an introduction form fault detection to fault tolerance*. 2006.

[21] Alan S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6) :601–611, 1976.

[22] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7) :603–614, 1984.

[23] H. Hammouri, M. Kinnaert, and E.H. El Yaagoubi. Observer-based approach to fault detection and isolation for nonlinear systems. *IEEE Transactions on Automatic Control*, 44(10) :1879–1884, 1999.

[24] P.M. Frank. Advanced fault detection and isolation schemes using nonlinear and robust observers. *IFAC Proceedings Volumes*, 20(5, Part 3) :63–68, 1987. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume III, Munich, Germany, 27-31 July.

[25] A. Zolghadri, D. Henry, and M. Monsion. Design of nonlinear observers for fault diagnosis : A case study. *Control Engineering Practice*, 4(11) :1535–1544, 1996.

[26] PAUL M. FRANK. On-line fault detection in uncertain nonlinear systems using diagnostic observers : a survey. *International Journal of Systems Science*, 25(12) :2129–2154, 1994.

[27] R.J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. *IFAC Proceedings Volumes*, 24(6) :65–81, 1991. IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'91), Baden-Baden, Germany, 10-13 September 1991.

[28] P.M. Frank. Analytical and qualitative model-based fault diagnosis – a survey and some new results. *European Journal of Control*, 2(1) :6–28, 1996.

[29] Silvio Simani, Cesare Fantuzzi, and Ron Patton. *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. 01 2003.

[30] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5) :709–719, 1997.

[31] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.

[32] Seongmin Heo and Jay H. Lee. Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, 51(18) :470–475, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[33] A. Soualhi, G. Clerc, H. Razik, and O. Ondel. Detection of induction motor faults by an improved artificial ant clustering. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pages 3446–3451, 2011.

[34] Joanna Sikorska, Melinda Hodkiewicz, and Lin Ma. Prognostic modeling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25 :1803–1836, 12 2011.

[35] Jianbo Liu, Dragan Djurdjanovic, K. Marko, and Jun Ni. Divide and conquer approach to anomaly detection, localization and diagnosis. *Mechanical Systems and Signal Processing*, 23 :2488–2499, 11 2009.

[36] George Vachtsevanos, F.L. Lewis, Michael Roemer, Andrew Hess, and Blake wu. Intelligent fault diagnosis and prognosis for engineering systems : Methods and case studies. 13, 01 2006.

[37] E. SHANNON. A mathematical theory of communication. *The Bell System Technical Journal*, 1948.

[38] Ted Petrie Leonard E. Baum. December, 1966 statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Statist*, 1966.

[39] R. L. Cave and L. P. Neuwirth. Hidden markov models for english. *in J. D. Ferguson*, October 1980.

[40] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.

[41] Diksha Khiatani and Udayan Ghose. Weather forecasting using hidden markov model. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pages 220–225, 2017.

[42] Ferdinando Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. volume 22, pages 138 – 142, 01 1995.

[43] Yassine Kasmi, Amina Eladdadi, Youssef Zeroual, Abdellatif Touzani, and Issam Meftah Kadmiri. A hidden markov model for predicting the methylome of the dna adenine methyltransferase in bacterial genome. *Gene Reports*, 27 :101633, 2022.

[44] Richard Kay. A markov model for analysing cancer markers and disease states in survival studies. *Biometrics*, 42(4) :855–865, 1986.

[45] Cristian López, Ángel Naranjo, Siliang Lu, and Keegan J. Moore. Hidden markov model based stochastic resonance and its application to bearing fault diagnosis. *Journal of Sound and Vibration*, 528 :116890, 2022.

[46] Houman Hanachi, Chris Mechefske, Jie Liu, Avisekh Banerjee, and Ying Chen. Performance-based gas turbine health monitoring, diagnostics, and prognostics : A survey. *IEEE Transactions on Reliability*, PP :1–24, 05 2018.

[47] Shuhui Wang, Jiawei Xiang, Yongteng Zhong, and Yuqing Zhou. Convolutional neural network-based hidden markov models for rolling element bearing fault identification. *Knowledge-Based Systems*, 144 :65–76, 2018.

[48] Maria Fox, Malik Ghallab, Guillaume Infantes, and Derek Long. Robot introspection through learned hidden markov models. *Artificial Intelligence*, 170 :59–113, 02 2006.

[49] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society : Series B*, 39 :1–38, 1977.

[50] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1) :164–171, 1970.

[51] Krogh A Mitchison G Durbin R, Eddy SR. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[52] Lawrence R. Rabiner and Biing-Hwang Juang. Hidden markov models for speech recognition — strengths and limitations. 1992.

[53] Abdenaïm EL YACOUBI. *Modélisation Markovienne de l'écriture manuscrite Application à la reconnaissance des adresses postales*. PhD thesis, L'UNIVERSITE DE RENNES 1, 1996.

[54] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4) :1035–1074, 1983.

[55] L.R. Rabiner, J.G. Wilpon, and F.K. Soong. High performance connected digit recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8) :1214–1225, 1989.

[56] Kai-Fu Lee, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Xuedong Huang. Speech recognition using hidden markov models : A cmu perspective. *Speech Communication*, 9(5) :497–508, 1990. Neuropeech '89.

[57] Amlan Kundu, Yang He, and Paramvir Bahl. Recognition of handwritten word : First and second order hidden markov model based approach. *Pattern Recognition*, 22 :283–297, 12 1989.

[58] Longbing Cao, Philip S. Yu, and Vipin Kumar. Nonoccurring behavior analytics : A new area. *IEEE Intelligent Systems*, 30(6) :4–11, 2015.

[59] Pascal VRIGNAT. *Génération d'indicateurs de maintenance par une approche semi-paramétrique et par une approche markovienne.* PhD thesis, UNIVERSITE D'ORLEANS, 2010.

[60] Claude E. Shannon. A mathematical theory of communication. *he Bell System Technical Journal*, 27 :379–423, 1948.

[61] F Bouquet. Méthodes et outils pour l'intelligence artificielle. Technical report, 2003.

[62] Luc Pronzato, Eric Thierry, and Eric Wolsztynski. Minimum entropy estimation in semi-parametric models : a candidate for adaptive estimation ? 01 2004.

[63] J. A. Cover, T. M. et Thomas. *Elements of information theory.* Wiley, 1991.

[64] Srinivasan S Iyengar David Hocker, Xiaohu Li. Shannon entropy based time-dependent deterministic sampling for efficient on-the-fly quantum dynamics and electronic structure. *Chem. Theory Comput*, page 256–268., 2011.

[65] Jan Beirlant, E. Dudewicz, L. Gyor, and E.C. Meulen. Nonparametric entropy estimation : An overview. *International Journal of Mathematical and Statistical Sciences*, 6, 01 1997.

[66] *Étude de la pertinence des paramètres stochastiques sur des Modèles de Markov Cachés*, 2013.

[67] Darío Iglesias, J. Bernal-Chaves, Carmen Peláez-Moreno, Ascensión Gallardo-Antolín, and Fernando Díaz-de María. A speech recognizer based on multiclass svms with hmm-guided segmentation. pages 257–266, 04 2005.

[68] Lingling Wang, Ying Zhou, Rao Li, and Lieyun Ding. A fusion of a deep neural network and a hidden markov model to recognize the multiclass abnormal behavior of elderly people. *Knowledge-Based Systems*, 252 :109351, 07 2022.

[69] Padhraic Smyth. Clustering sequences with hidden markov models. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, page 648–654, Cambridge, MA, USA, 1996. MIT Press.

[70] Longbing Cao, Xiangjun Dong, and Zhigang Zheng. e-nsp : Efficient negative sequential pattern mining. *Artificial Intelligence*, 235 :156–182, 2016.

[71] Zhigang Zheng, Yanchang Zhao, Ziye Zuo, and Longbing Cao. An efficient ga-based algorithm for mining negative sequential patterns. In Mohammed J. Zaki, Jeffrey Xu Yu, B. Ravindran, and Vikram Pudi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 262–273, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[72] Ping Qiu, Yongshun Gong, Yuhai Zhao, Longbing Cao, Chengqi Zhang, and Xiangjun Dong. An efficient method for modeling nonoccurring behaviors by negative sequential patterns with loose constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 34(4) :1864–1878, 2023.

[73] Sue-Chen Hsueh, Ming-Yen Lin, and Chien-Liang Chen. Mining negative sequential patterns for e-commerce recommendations. In *2008 IEEE Asia-Pacific Services Computing Conference*, pages 1213–1218, 2008.

[74] J. Lember and A. Koloydenko. Adjusted viterbi training, 2004.

[75] Luis Javier Rodriguez-Fuentes and M. Torres. Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition. pages 847–857, 06 2003.

[76] Nir Shlezinger, Yonina C. Eldar, Nariman Farsad, and Andrea J. Goldsmith. *ViterbiNet : Symbol Detection Using a Deep Learning Based Viterbi Algorithm.* 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2019.

[77] Zied Baklouti. Tweet sentiment extraction using viterbi algorithm with transfer learning. August 2023.

[78] Abdelkrim Bekkhoucha Imad Sassi, Samir Anter. Fast parallel constrained viterbi algorithm for big data with applications to financial time series. page Pages 50–55, 09 September 2021.