



**HAL**  
open science

# Découverte et généralisation de connaissances appliquées aux procédés de fabrication

Mahmoud El Krim Ferhat

► **To cite this version:**

Mahmoud El Krim Ferhat. Découverte et généralisation de connaissances appliquées aux procédés de fabrication. Informatique. Nantes Université, 2022. Français. NNT : 2022NANU4044 . tel-03966128

**HAL Id: tel-03966128**

**<https://theses.hal.science/tel-03966128>**

Submitted on 31 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE N° 601  
*Mathématiques et Sciences et Technologies*  
*de l'Information et de la Communication*  
Spécialité : *Informatique*

Par

**Mahmoud El Krim FERHAT**

**Decouverte et Généralisation de Connaissances appliquées aux  
Procédés de Fabrication**

Thèse présentée et soutenue à « l'amphi2 ISITEM, Nantes », le « 05/12/2022 »  
Unité de recherche : **Laboratoire des Sciences du Numérique de Nantes**

## **Rapporteurs avant soutenance :**

Julien LE DUIGOU      Maître de Conférences (HDR) à l'Université de Technologie de Compiègne  
Faïcel CHAMROUKHKI      Professeur des Universités à l'Université de Caen Normandie

## **Composition du Jury :**

Président :	Benoit IUNG	Professeur des Universités à l'Université de Lorraine
Examineurs :	Faïcel CHAMROUKHKI	Professeur des Universités à l'Université de Caen (détaché à l'IRT SystemX)
	Julien LE DUIGOU	Maître de Conférences (HDR) à l'Université de Technologie de Compiègne
	Catherine DA CUNHA	Professeure des Universités à Centrale Nantes
	Benoit IUNG	Professeur des Universités à l'Université de Lorraine
Dir. de thèse :	Philippe LERAY	Professeur des Universités à l'Université de Nantes
Co-enc. de thèse :	Mathieu RITOU	Maître de Conférences (HDR) à l'Université de Nantes

## **Invité(s) :**

Nicolas LE DU      Établissement Georges Renault



# ACKNOWLEDGEMENT

---



Avant tout, je tiens à remercier Dieu, le Tout Puissant, pour m'avoir donné la force de résister et de persévérer même durant les périodes les plus difficiles.

Je souhaite ensuite remercier les personnes les plus chères à mes yeux, mes parents, qui m'ont élevé, instruit, toujours cru en moi, et m'ont soutenu durant toutes les étapes de ma vie. L'homme que je suis devenu n'aurait certainement pas existé sans votre présence. Je suis très chanceux et fier d'avoir des parents comme vous. Je remercie également ma soeur Amel et mon frère Mehdi pour tous les souvenirs et les bons moments qui m'ont permis de tenir pendant toutes ces années.

Je tiens également à remercier ma femme Asmaa pour son soutien durant toute la thèse. Ta patience et ta compréhension ont été décisives pour l'aboutissement de ce travail. Je te remercie infiniment, que Dieu te garde pour moi.

J'adresse mes remerciements à ma tante Assia qui m'a soutenu par sa présence, sa gentillesse, et son hospitalité durant tout mon séjour en France.

Je remercie très fortement mes encadrants, M. Philippe Leray et M. Mathieu Ritou, M. Nicolas Le Du, M. Laurent Macquet, qui m'ont accompagné durant ces trois années, pour leurs conseils, la confiance et l'autonomie qu'ils m'ont accordés, et leur disponibilité précieuse qui m'a permis de toujours avoir un appui lorsque j'en avais besoin.

J'adresse également toute ma reconnaissance aux personnes d'entreprise Desoutter et plus particulièrement l'équipe de la Concept Team (Pierre, Olivier, Stéphane, Justine), pour les discussions passionnantes en réunion, aux détours de couloirs ou devant la machine à café, ainsi que pour l'ambiance de travail décontractée et chaleureuse.



# TABLE OF CONTENTS

---

<b>Introduction</b>	<b>9</b>
Contexte du travail et motivations . . . . .	9
Plan de la thèse . . . . .	14
État de l'art . . . . .	14
Contributions . . . . .	15
<b>I État de l'art</b>	<b>17</b>
<b>1 Détection de défauts et diagnostic dans les systèmes industriels</b>	<b>18</b>
1.1 Définitions de base . . . . .	18
1.1.1 Types de défauts . . . . .	18
1.1.2 Aperçu du processus de diagnostic des défauts . . . . .	20
1.2 Méthodes de surveillance process . . . . .	23
1.2.1 Méthodes traditionnelles . . . . .	23
1.2.2 Méthodes basées sur la connaissance . . . . .	24
1.2.3 Méthodes basées-modèle . . . . .	25
1.3 Méthodes basées sur les données . . . . .	26
1.3.1 Approches supervisées . . . . .	27
1.3.2 Approches non-supervisés . . . . .	29
1.3.3 Discussion . . . . .	30
1.4 Conclusion . . . . .	32
<b>2 Modèle à mélange de gaussiennes</b>	<b>33</b>
2.1 Modèle de Mélange de Gaussiennes . . . . .	33
2.1.1 GMM . . . . .	33
2.1.2 Maximum de vraisemblance . . . . .	35
2.1.3 L'algorithme d'Espérance-Maximisation (EM) . . . . .	37
2.1.4 Limitations de l'algorithme EM . . . . .	38
2.1.5 Critères de sélection de modèle . . . . .	39

## TABLE OF CONTENTS

---

2.1.6	Lien entre gaussiennes et clusters . . . . .	41
2.2	Décisions avec rejet . . . . .	42
2.2.1	Rejet de distance . . . . .	43
2.2.2	Rejet d'ambiguïté . . . . .	45
2.2.3	Conclusion . . . . .	47
2.3	Réduction de modèles . . . . .	47
2.3.1	Formalisation du problème . . . . .	47
2.3.2	Mesures de dissimilarité . . . . .	48
2.3.3	Réduction gloutonne . . . . .	51
2.3.4	Affinement . . . . .	54
2.3.5	GMRC . . . . .	55
2.4	Conclusion . . . . .	58
<b>3</b>	<b>Ontologie</b>	<b>61</b>
3.1	Notions de base . . . . .	62
3.1.1	Définition . . . . .	62
3.1.2	Les catégories de l'ontologie . . . . .	63
3.1.3	Les utilisations d'ontologie . . . . .	64
3.1.4	Évolution de l'ontologie . . . . .	65
3.2	Ontologie pour le diagnostic de défauts . . . . .	66
3.2.1	Partage du vocabulaire . . . . .	67
3.2.2	Raisonnement sur les causes de défauts . . . . .	67
3.2.3	Calcul des similarités . . . . .	68
3.2.4	Synthèse . . . . .	69
3.3	Approches hybrides . . . . .	69
3.3.1	L'apprentissage automatique pour l'ontologie . . . . .	70
3.3.2	L'ontologie pour l'apprentissage automatique . . . . .	70
3.3.3	Quelques critiques des approches citées . . . . .	71
3.4	Conclusion . . . . .	72
<b>II</b>	<b>Contributions</b>	<b>75</b>
<b>4</b>	<b>Découverte itérative de connaissances pour la détection de défauts</b>	<b>76</b>
4.1	L'architecture proposée . . . . .	76

4.1.1	Architecture globale de détection de défaut inconnu . . . . .	77
4.1.2	Peuplement de l'ODD . . . . .	79
4.1.3	Enrichissement de l'ODD . . . . .	82
4.2	Expérimentations . . . . .	85
4.2.1	Protocole Expérimental . . . . .	85
4.2.2	Détection et apprentissage de nouveaux défauts . . . . .	87
4.2.3	Robustesse . . . . .	90
4.3	Conclusion . . . . .	92
<b>5</b>	<b>Généralisation et exploitation des connaissances</b>	<b>95</b>
5.1	Formalisation du problème . . . . .	95
5.2	Description de la solution . . . . .	96
5.2.1	Évolution de la structure l'ontologie . . . . .	96
5.2.2	Mécanisme 1 : Réduction intra-contexte . . . . .	97
5.2.3	Mécanisme 2 : Généralisation inter-contextes . . . . .	99
5.2.4	Algorithme de réduction de modèle gaussien . . . . .	101
5.3	Expérimentations . . . . .	105
5.3.1	Protocole expérimental . . . . .	105
5.3.2	Mécanisme 1 : Réduction intra-contexte . . . . .	106
5.3.3	Mécanisme 2 : Généralisation inter-contextes . . . . .	111
5.3.4	Discussions et pistes d'amélioration . . . . .	116
5.4	Conclusion . . . . .	119
<b>6</b>	<b>Decouverte et généralisation de connaissances en vissage</b>	<b>123</b>
6.1	Présentation du métier . . . . .	123
6.1.1	Notions de base sur les assemblages boulonnés . . . . .	123
6.1.2	Détection de défauts et surveillance . . . . .	125
6.2	Découverte incrémentale de nouveaux défauts . . . . .	127
6.2.1	Protocole expérimental . . . . .	127
6.2.2	Peuplement de l'ontologie . . . . .	128
6.2.3	Classification et enrichissement . . . . .	129
6.2.4	Conclusion . . . . .	130
6.3	Généralisation des connaissances . . . . .	131
6.3.1	Protocole expérimental . . . . .	131
6.3.2	Réduction intra-contextes . . . . .	132



## TABLE OF CONTENTS

---

6.3.3	Généralisation inter-contextes . . . . .	135
6.3.4	Conclusion . . . . .	140
6.4	Industrialisation de l'architecture . . . . .	140
6.5	Conclusion . . . . .	141
<b>Conclusion</b>		<b>145</b>
	Perspectives . . . . .	148
<b>Bibliographie</b>		<b>153</b>
<b>Annexe</b>		<b>177</b>
	Résultats de la généralisation (chapitre 5) . . . . .	177
	Courbes de vissage . . . . .	180

# INTRODUCTION

---

## Contexte du travail et motivations

Le développement de la technologie a accompagné les industries des systèmes mécaniques vers des usines hautement automatisées et intelligentes dans lesquelles les machines intelligentes et les réseaux de capteurs interagissent mutuellement. Avec le développement de ces technologies, un nouveau concept, l'industrie 4.0, a été introduit par les allemands et symbolisant le début de la 4<sup>e</sup> révolution industrielle [129]. Les trois premières révolutions industrielles ont entraîné des changements radicaux dans la fabrication, à savoir la mécanisation, l'utilisation de l'électricité et de l'automatisation (l'électronique). L'industrie 4.0 repose sur l'exploitation des données numériques de fabrication et s'accompagne de trois améliorations technologiques, l'Internet des Objets Industriels (IIoT), l'Intelligence Artificielle (IA), et les systèmes cyber-physiques (CPS) (et plus récemment le Jumeau Numérique (DT)) qui ont progressé au cours de la dernière décennie.

Les CPS sont définis comme des systèmes physiques (typiquement un système de production) et une représentation numérique de son état, grâce à des capteurs installés sur le système de production [91]. Les CPS n'étaient pas nécessairement interconnectés avant l'industrie 4.0, mais doivent l'être actuellement. L'IIoT est un réseau d'appareils industriels connectés qui peuvent communiquer entre eux et fournir des données aux utilisateurs par le biais d'Internet. Le déploiement de l'IIoT enrichit les sources d'informations disponibles dans l'atelier. En effet, grâce à leur connectivité aisée (de type plug&play), ils permettent de mettre à disposition facilement une information. Par conséquent, l'utilisation croissante de capteurs et de machines en réseau a entraîné la génération continue de données complexes, à haut volume et à grande vitesse, également appelées Big Data [62].

En effet, de grands volumes de données collectés dans les industries sont partiellement stockés et restent sous-exploités. L'analyse de ces données devient plus importante car c'est un levier majeur d'amélioration de la performance industrielle, permise par les développements rapides sur le domaine des données. En revanche, une analyse manuelle de ces données devient impossible du fait du grand volume de données hétérogènes. Par conséquent, l'analyse de données automatique par l'Intelligence Artificielle (IA) est néces-

saire [140]. En effet, l'analyse de toutes ces données peut apporter un avantage concurrentiel important aux entreprises, car elles sont en mesure d'évaluer de manière significative l'ensemble des processus [116]. Ces processus peuvent se trouver à une échelle globale telle que la gestion de la chaîne logistique, gestion des relations clients, ou bien à une échelle réduite comme l'ordonnancement des ordres de fabrication, le contrôle qualité, et le diagnostic et la détection de défauts dans les procédés de fabrication et d'assemblage [70].

Dans le cadre de l'analyse de données, la détection des défauts dans les procédés de fabrication joue un rôle clé dans la traçabilité et l'optimisation de la production. Elle s'appuie généralement sur la surveillance en cours de process, pour laquelle une littérature abondante est disponible. Globalement, il existe 4 approches de détection de défauts : méthodes traditionnelles, méthodes basées sur des modèles physiques, méthodes basées sur la connaissance, méthodes basées sur les données. La dernière approche a suscité beaucoup plus d'intérêt ces dernières années en raison de l'augmentation de la puissance de calcul ainsi que la disponibilité des données. Par exemple, la classification et le clustering ont été appliqués avec succès au processus d'usinage intelligent [83]. De même, de nombreuses approches de détection de défauts ont été développées pour les opérations de vissage [4].

Dans la littérature, 79% des méthodes de détection de défauts basées sur les données proviennent des méthodes basées sur l'apprentissage automatique (ML) supervisées [114]. C'est à dire que ces méthodes requièrent la connaissance de la vérité terrain afin d'établir un modèle mathématique pour la surveillance process. Plusieurs techniques ont été proposées, telles que les réseaux neuronaux convolutifs (CNN) [113], les systèmes flous [45], ou les machines à vecteurs de support [104]. Une limitation générale est que la connaissance du domaine est rarement intégrée dans les approches. De plus, plusieurs modèles d'apprentissage peuvent être proposés et il n'existe pas de consensus sur la meilleure architecture à adopter pour une application donnée. En outre, les ensembles de données d'entraînement sont généralement issus d'expériences menées dans des conditions de laboratoire idéales, où une personne peut manuellement étiqueter chacune des quelques opérations comme correcte ou incorrecte. À l'inverse, connaître la vérité terrain dans un contexte industriel, avec un rythme de production intensif, n'est généralement pas possible. C'est la raison pour laquelle les méthodes d'apprentissage non-supervisées sont souvent nécessaire sur les applications industrielles.

De plus, même si des étiquettes pouvaient être fournies, ces méthodes sont généralement entraînées sur un nombre limité de défauts connus, avec peu ou pas d'observation de ces défauts. Dans le cas où le processus est mature et a été étudié en profondeur, ses

principales défaillances sont bien connues et peuvent être surveillées avec succès. Mais pour les processus moins étudiés, la détection de défauts inconnus est un problème important. De plus, en plus de la capacité de détection de défauts inconnus, il est souhaitable d'intégrer les connaissances associées aux défauts nouvellement détectés dans le modèle de surveillance. Ainsi, des défauts inconnus peuvent potentiellement survenir en production et, même connus, les défauts sont souvent manquants dans les jeux de données industrielles d'entraînement. Ceci est expliqué par le fait que la fiabilité sur une production industrielle doit être de 100%, et donc, même si les défauts sont connus, les jeux de données résultants sont très déséquilibrés (même pour un grand volume de données) et il est difficile (voir impossible) d'apprendre les défauts. C'est pourquoi des méthodes permettant la découverte de nouveaux défauts sont cruciales. Pourtant elles sont rarement discutées explicitement dans la littérature.

Enfin, il convient de noter que de nombreux travaux de recherche se concentrent sur un aspect unique, une stratégie de processus avec un seul jeu de paramètres opératoires (i.e. contexte), ce qui n'est pas suffisant pour les applications pratiques où une multitude de paramètres existent et affectent la signature des défauts. En effet, un changement mineur sur un paramètre (i.e. changement de contexte) peut nuire à la fiabilité du modèle de surveillance. Cela est expliqué par le fait que l'hypothèse de *l'indépendance et l'identité distribuées* (i.e. *iid*) des données sur laquelle repose la plupart des modèles d'apprentissage automatique n'est potentiellement plus respectée. Pour y remédier, une stratégie intuitive consiste alors à apprendre un nouveau modèle de surveillance sur les données issues du nouveau contexte de production. Cette approche requière néanmoins un nombre suffisant de données pour apprendre un modèle de surveillance. De plus, toutes les connaissances apprises sur les anciens contextes (i.e. en termes de défauts et de modèles appris) deviennent obsolètes, ce qui affecte négativement l'efficacité de la surveillance process (absence d'exploitation des connaissances apprises). Par conséquent, il convient de concevoir des mécanismes permettant d'exploiter les connaissances antérieures sur de nouveaux contextes. À cet effet, des travaux de l'Adaptation au Domaine et de Transfert Learning sont généralement proposés pour adresser ces problèmes. En revanche, ces derniers se reposent sur des réseaux de neurones profonds (boîte-noire) [164, 166] qui nécessitent un volume de données important. De plus, ces derniers nécessitent dans la plupart des jeux d'apprentissage (domaine source) où la vérité terrain est connue, ce qui est rarement le cas en industrie.

Autre alternative, une ontologie peut définir explicitement les concepts et leurs rela-

tions dans un domaine d'expertise [67]. Elle permet également le raisonnement et l'interrogation des connaissances à l'aide de règles et de moteurs d'inférence. Elles constituent donc des méthodes prometteuses pour la modélisation, la gestion et l'utilisation avancée de multitude de connaissances. La détection des défauts basée sur la gestion des connaissances (*Knowledge Management* (KM)) se concentre sur la représentation et l'utilisation de la connaissance des défauts. Avec le développement de la technologie du web sémantique, l'ontologie pour le diagnostic des défauts a été étudié et appliqué dans divers domaines [173]. L'utilisation de l'ontologie pour le diagnostic des défaillances peut être divisée en trois catégories [99] : le partage du vocabulaire du domaine, le raisonnement pour l'identification de la cause de la défaillance et la récupération des informations ou la reconnaissance des similarités. Néanmoins, ces méthodes n'utilisent généralement pas les données issues des opérations de fabrication qui pourraient améliorer de manière significative les performances (en termes de précision du diagnostic, par exemple) [169]. Ainsi, l'extraction de connaissances par apprentissage automatique est pertinente. Elle permet de modéliser les défauts et de créer à terme une ontologie de diagnostic de défaut.

En outre, l'utilisation des ontologies pour améliorer les performances des techniques de ML a été rarement discuté dans la littérature. En effet, la tendance a été plus centrée sur l'utilisation des techniques de ML pour alimenter une ontologie pour des fins de diagnostic de défauts. En revanche, l'exploitation des connaissances de l'ontologie peut améliorer les performances des techniques de ML. À titre d'exemple, il serait intéressant d'exploiter les connaissances apprises de la surveillance d'une gamme de produit (en termes de défauts et de modèles de surveillance), puis de transférer ces connaissances lors de la conception d'un nouveau produit. Toutes les connaissances apprises antérieurement serviront donc d'un point de départ pour enclencher la surveillance sur le nouveau composant. Ce raisonnement est analogue au Transfert Learning et à l'Adaptation au Domaine qui consiste à utiliser un modèle de ML entraîné sur une tâche (source) pour être transféré vers une autre tâche (cible) similaire. Les connaissances du modèle (paramètres et hyperparamètres d'un réseau de neurones profond) seront donc utilisées pour apprendre un modèle performant sur la tâche cible.

L'objectif de cette thèse est dans un premier temps de proposer une architecture hybride (ML-KM) dédiée à la détection de défauts potentiellement inconnus dans les procédés de fabrication. L'architecture proposée est évolutive dans le sens où de nouveaux défauts qui n'étaient pas appris au début de la surveillance process peuvent être détectés et intégrés itérativement dans l'architecture. À cet effet, un mécanisme d'IA permettant

la détection de défaut inconnu est fortement recherché. Ensuite, une ontologie alimentée par des techniques d'apprentissage automatique (ML) est proposée. De plus, l'intégration des connaissances est assurée par l'intervention judicieuse d'un expert du domaine afin de labéliser les clusters de défauts découverts (plutôt qu'un laborieux étiquetage de milliers de données). Le second objectif est d'exploiter les connaissances de l'ontologie pour améliorer les performances des techniques d'apprentissage automatique. En effet, l'ajout itératif des connaissances dans l'ontologie donne l'opportunité d'exploiter ces dernières au fil du temps, ce qui permet d'affiner les modèles de ML utilisées.

De ce fait, l'architecture améliore les performances de chacune des composantes de façon cyclique. En d'autres termes, l'apprentissage automatique alimente et enrichit les connaissances de l'ontologie. Puis, l'exploitation des connaissances intégrées itérativement dans l'ontologie améliorent les performances des techniques de ML utilisées. Enfin, une architecture générique a été recherchée dans le sens où elle est capable de traiter des données de fabrication de natures différentes. À cet effet, elle a été testée sur des données UCI ainsi que sur des données réelles provenant des opérations de vissage des clients de la société CP Georges Renault.

En effet, la thèse est réalisée dans le cadre d'une convention CIFRE au sein de l'entreprise CP Georges Renault qui est spécialisée dans le développement d'outillage pour l'industrie. Elle fait partie du groupe suédois Atlas Copco présent et reconnu sur les 5 continents. Ces produits sont commercialisés sous différentes marques, dont Desoutter et Chicago Pneumatic, pour différents secteurs de l'industrie (automobile, aéronautique, [www.desouttertools.fr](http://www.desouttertools.fr)). Son Technocentre, situé à Nantes, est chargé de développer des systèmes de vissage et de perçage à forte valeur ajoutée. La technologie des visseuses est déjà très avancée, car ce sont de véritables objets industriels connectés (IIoT), communiquant avec le système d'information de l'entreprise. Les développements les plus poussés se situent dans l'industrie automobile. Dans son challenge Desoutter 4.0, l'entreprise souhaite passer un nouveau cap et proposer à ses clients de nouvelles fonctions d'intelligence artificielle pour ses visseuses connectées. L'objectif industriel est d'être en mesure de détecter des défauts inconnus lors des opérations de vissage et de garantir une haute fiabilité dans les prédictions de la qualité des serrages effectués.

## **Plan de la thèse**

La thèse est divisée en deux parties, la partie I traite de l'état de l'art et la partie II détaille les contributions scientifiques de la thèse.

### **État de l'art**

Cette partie est divisée en 3 chapitres présentant l'état de l'art associé à la détection de défauts dans la fabrication industrielle. Il convient de noter que les limites sont précisées et discutées dans chaque chapitre. Ensuite, ces limites peuvent être adressées partiellement par les techniques et outils présentés dans le chapitre suivant.

Le premier chapitre présente une revue générale sur la détection et diagnostic de défauts dans les systèmes industriels, et plus particulièrement les installations de production industrielle. Pour ce faire, les notions de bases telles que les types de défauts ainsi que les étapes du diagnostic de défauts seront présentées. Ensuite, une revue des techniques de détection de défauts est fournie, en mettant l'accent sur les techniques basées sur les données. Enfin, les limites de ces techniques sont présentées et une focalisation sur les modèles de mélange de gaussiennes (GMM) est choisie.

Le deuxième chapitre est consacré à la présentation des notions et techniques associées aux GMM. Pour ce faire, les notions de base liées à ces modèles sont présentées. En effet, les techniques d'apprentissage, la sélection de modèles et le lien entre les classes de défauts et clusters seront détaillés. Ensuite, les mécanismes de décision avec rejet permettant de détecter des événements inconnus seront détaillés. Ces derniers peuvent correspondre à de nouveaux défauts ou à une dérive d'un défaut connu. Ainsi, ces techniques seront fortement souhaitées pour la détection de nouveaux défauts. Enfin, la réduction de modèles de GMM (GMR) permettant de réduire le nombre de gaussiennes dans un GMM sera décrite. Ces techniques permettent d'éliminer des redondances ou obsolescences de gaussiennes et de réduire le sur-apprentissage. Par conséquent, elles sont utiles pour la surveillance process en temps réel. Enfin, les limites des GMM appliqués à la surveillance process seront présentées.

Le troisième chapitre est consacré aux ontologies. À cet effet, les notions de bases (concepts, utilisations, évolutions) liées à ce paradigme seront illustrées. Ensuite, les applications d'ontologie pour le diagnostic de défauts (détection de défaut y compris) seront

détaillées. Enfin, nous présentons des approches hybrides ayant combiné l'apprentissage automatique avec la gestion des connaissances, en mettant en avant leurs limites. Ces dernières limites ainsi que celles énoncées sur les 2 premiers chapitres serviront de point de départ pour les contributions de cette thèse.

## **Contributions**

La partie contributions de la thèse est constituée de trois chapitres qui englobent deux thèmes majeurs de recherche et leurs applications.

Dans le chapitre 4, nous construisons la fondation de notre architecture. Cette dernière est dédiée à la découverte des défauts de façon itérative sous forme de connaissances. Ainsi, cette partie se base sur une association innovante des techniques d'apprentissage automatique couplées à la connaissance d'expert pour alimenter une ontologie. Pour ce faire, les différentes étapes constituant cette architecture seront présentées. Puis, pour chaque brique fonctionnelle, les techniques utilisées ainsi que leurs intérêts seront détaillées. Enfin, les propriétés et performance de notre architecture, en la testant sur des données UCI en définissant un protocole expérimental spécifique à chaque propriété seront montrées.

Le chapitre 5 présente l'exploitation des connaissances de l'ontologie ajoutées itérativement par l'architecture proposée. Pour ce faire, nous proposons des mécanismes de généralisation de connaissances présentes dans une ontologie en exploitant la hiérarchie des concepts. Cette généralisation permet d'améliorer les performances des modèles de ML pour la surveillance process. Pour ce faire, la description du problème et des besoins seront présentés. Ensuite, des mécanismes de généralisation permettant d'adresser les problèmes formulés seront proposés. En outre, des techniques de GMR seront présentées pour la généralisation. Enfin, les mécanismes de généralisation seront expérimentés selon les protocoles définis dans le chapitre 4, ainsi que sur de nouveaux protocoles. À l'issue de l'expérimentation, l'autre partie de la boucle (ML-KM) initiée dans le chapitre 4 est complétée à travers les mécanismes de généralisation.

Comme le montre la figure 1, les contributions des deux premiers chapitres sont alors reprises dans le chapitre 6 qui détaille la découverte et la généralisation des connaissances appliquées sur des jeux de données enregistrés pendant des opérations de vissage, lors de réelles productions industrielles. Pour ce faire, une présentation du métier ainsi qu'une revue de littérature sur les techniques de détection de défauts en vissage sera présentée.



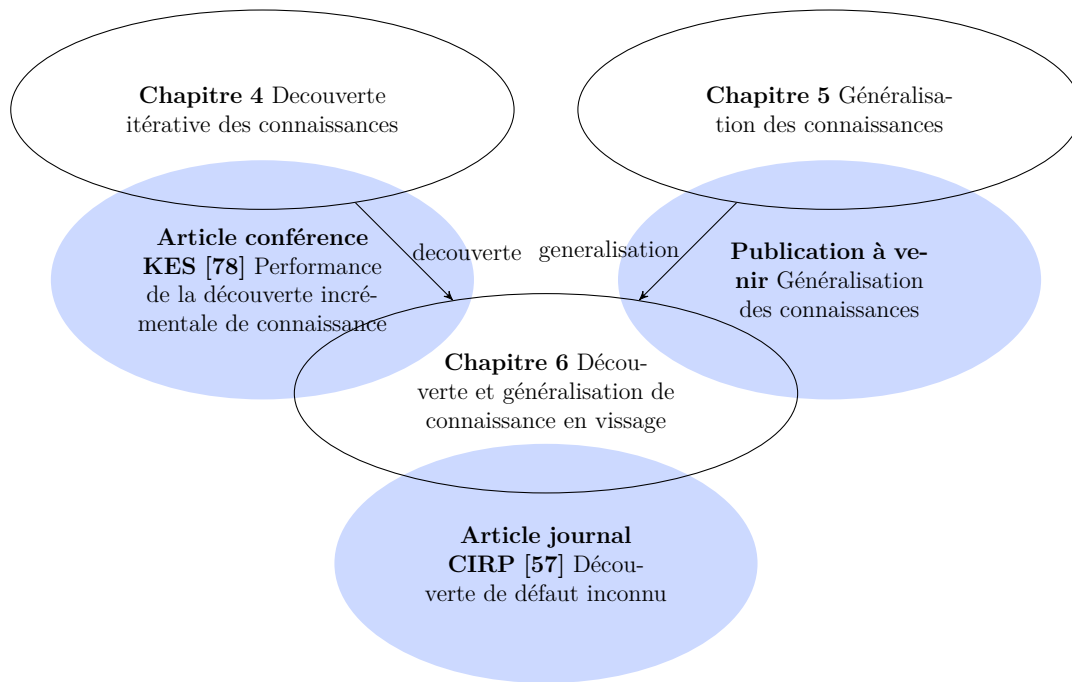


FIGURE 1 – Organisation des contributions du manuscrit : chapitres et publications

Ensuite, les expérimentation des techniques présentées dans les 2 premiers chapitres seront appliquées.

La structure générique détaillée dans le chapitre 4 a été originellement publiée dans l'article [78] et présentée lors de la conférence *26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2022)*. Le mécanisme de découverte de défauts inconnus et certains des résultats expérimentaux du chapitre 6 se retrouvent dans l'article [57] présenté dans le journal *CIRP Annals 2021* et d'autres figures certainement dans une publication à venir. Les derniers résultats obtenus dans cette thèse (chapitre 5 et certains résultats du chapitre 6) font l'objet d'une publication longue en cours de rédaction pour une soumission prochaine (article de journal).

PREMIÈRE PARTIE

# État de l'art

---

# DÉTECTION DE DÉFAUTS ET DIAGNOSTIC DANS LES SYSTÈMES INDUSTRIELS

---

Ce premier chapitre est dédié à la présentation de la détection et diagnostic de défauts dans les systèmes industriels, et plus particulièrement les installations de production. Pour ce faire, les notions de base liées à cette tâche sont présentées dans la section 1.1. Ensuite, un aperçu global sur les méthodes de détection défauts est illustré dans la section 1.2, en mettant en avant les méthodes basées sur les données. Enfin, une description plus détaillée sur les approches basées sur les données ainsi que leurs limites est fournie dans la section 1.3.

## 1.1 Définitions de base

Les défauts rencontrés dans les systèmes de production peuvent se produire dans l'installation elle-même (machines par exemple) ou dans ses instruments de mesure et de contrôle. Dans cette section, les types de défauts et les tâches de détection et de diagnostic de défauts seront décrits.

### 1.1.1 Types de défauts

En général, les défauts sont des déviations du comportement normal de l'installation (machine dans une usine de production) ou de son instrumentation (capteurs). Les défauts d'intérêt appartiennent à l'une des catégories suivantes [63] :

- **Défaut de composant** : désigne l'anomalie de certains composants ou même de sous-systèmes de l'objet contrôlé, qui rend l'ensemble du système incapable d'assurer normalement les fonctions établies.

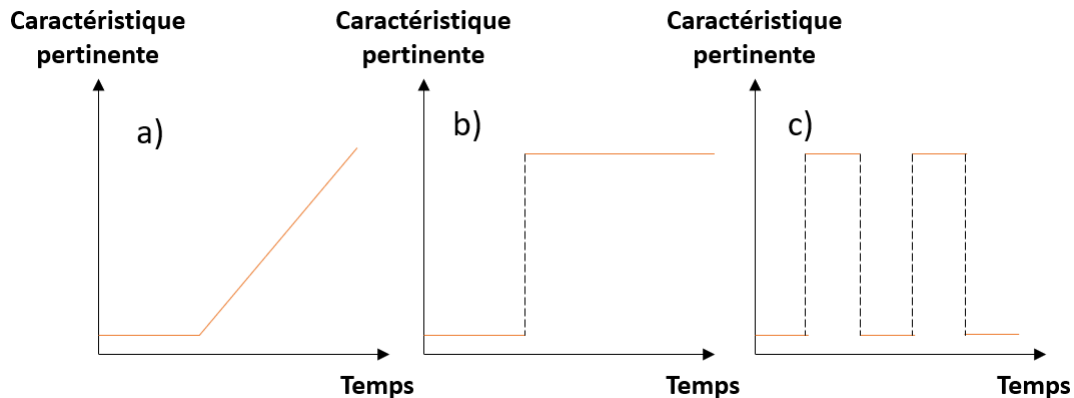


FIGURE 1.1 – Types de défauts selon l'aspect temporel : a) défaut de changement lent, b) défaut de changement soudain, c) défaut de intermittent [111]

- **Défaut de capteur** : se réfère au capteur utilisé pour la mesure de la grandeur physique déterminée. Un défaut dans ce dernier diminue la précision de la mesure, ce qui augmente l'écart entre la valeur mesurée et la valeur réelle.
- **Défaut process** : désigne l'anomalie associée à une opération de production sur une machine donnée. Elle peut être "générée" par un défaut d'opérateur, une mauvaise disposition de la pièce à produire, de mauvais paramètres opératoires, etc...
- **Défaut d'actionneur** : se réfère à l'actionneur qui est dans une condition défectueuse, ce qui résulte que la commande de contrôle ne peut pas être exécutée correctement. En d'autres mots, Il y a une différence entre la commande d'entrée et la sortie réelle de l'actionneur.

Selon les caractéristiques temporelles (figure 1.1), les défauts peuvent être classés en :

- **Défaut de changement lent** : défaut dans lequel le paramètre change lentement avec le temps ou l'environnement.
- **Défaut de changement soudain** : défaut dans lequel la valeur du paramètre mesurée par le capteur dévie soudainement et fortement.
- **Défaut de intermittent** : un défaut qui se produit de temps en temps en raison du vieillissement, d'une tolérance insuffisante ou d'un mauvais contact.

### 1.1.2 Aperçu du processus de diagnostic des défauts

Le diagnostic des défauts est un moyen d'évaluer l'état de défaut du système en utilisant toutes sortes d'informations existantes dans le système. Dans le processus de diagnostic des défauts, diverses méthodes d'inspection et de test peuvent être utilisées pour détecter la présence d'un défaut. Ensuite, l'emplacement et la cause du défaut peuvent être déterminés dans le système. Enfin, des mesures de correction sont appliquées afin d'éliminer le défaut détecté. Globalement, le processus de diagnostic des défauts comprend quatre parties (figure 1.2) : la détection des défauts, l'isolation des défauts, l'identification de la cause des défauts, et la réparation des défauts. Par ailleurs, chaque fois qu'un défaut est détecté, les procédures d'identification des défauts, de diagnostic des défauts et de récupération du processus sont utilisées dans l'ordre respectif ; sinon, seule la procédure de détection des défauts est répétée.

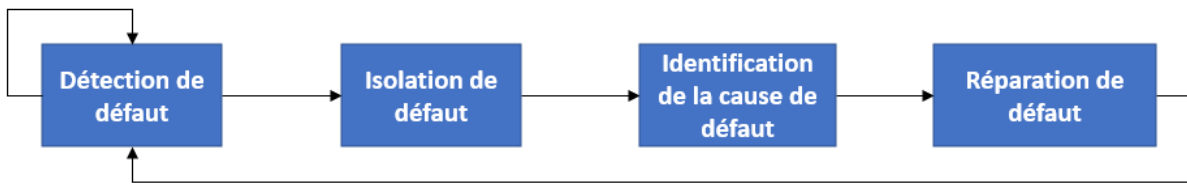


FIGURE 1.2 – Étapes de diagnostic de défauts [93]

**Détection des défauts** La détection des défauts fait référence au processus consistant à détecter si un défaut se produit dans un système et à déterminer le moment où le défaut se produit. La détection des défauts est généralement la première étape du processus de diagnostic des défauts et constitue la prémisse de l'isolation et de l'identification de la cause des défauts qui suivent. Si aucun défaut n'est détecté, cela indique que le système est dans un état de fonctionnement normal. Si un défaut est détecté, des moyens appropriés peuvent être sélectionnés pour analyser plus en profondeur l'emplacement, le type et la valeur du défaut.

Dans les premières recherches sur le diagnostic des défauts, seule la partie détection des défauts était réalisée. Plus tard, avec le développement de la technologie de diagnostic des défauts, l'analyse ultérieure des défauts a été progressivement développée et appliquée. Les méthodes de détection de défauts seront détaillées en profondeur dans la section 1.2.

Enfin, il existe généralement trois indices pour juger de la performance de la détection

des défauts [63] :

- **Sensibilité** : Fait référence à la capacité d'un système à détecter de petits signaux de défauts. Plus la sensibilité du système de détection est élevée, plus le signal de défaut qu'il peut détecter est petit.
- **Rapidité** : La capacité du système de détection à détecter le défaut le plus rapidement possible après son apparition. Plus la détection des défauts est rapide, plus l'intervalle de temps entre l'apparition du défaut et sa détection est court.
- **Taux de fausses alarmes et taux de détection manquante** : Les fausses alarmes correspondent à une situation où le système ne présente pas de défaut mais a été déterminé à tort comme présentant un défaut. La détection manquante fait référence à la situation où un défaut s'est produit dans le système mais n'a pas été détecté. Un système de détection de défauts fiable doit maintenir le taux de fausses alarmes et le taux de détection manquante aussi bas que possible.

**Isolation des défauts** En maintenance, l'isolation des défauts consiste à localiser le défaut du système après avoir déterminé que le système est défaillant. L'isolation du défaut est utilisée pour déterminer l'état de fonctionnement du sous-système ou de l'équipement qui est tombé en panne. Puis, pour localiser l'emplacement spécifique du défaut et pour ouvrir la voie à l'analyse du type ou de la sévérité du défaut.

La capacité d'isolation des défauts dépend de la capacité du système de diagnostic à distinguer différents défauts. Cette dernière dépend des caractéristiques physiques de l'objet surveillé, de la taille du défaut, du bruit, des interférences, des erreurs de modélisation et de l'algorithme de diagnostic conçu. Plus la capacité de séparation est forte, plus la localisation du défaut est précise.

Dans le cadre de la détection et l'isolation de défauts inter-tours d'un générateur de centrale électrique, Michau et al [109] ont proposé une méthode basée sur les machines d'apprentissage extrêmes hiérarchiques (HELM) pour traiter de grands ensembles de données comprenant des signaux de natures différentes, des signaux non informatifs, des relations non linéaires et du bruit. La méthode comprend des auto-encodeurs empilés qui sont capables d'apprendre les caractéristiques de haut niveau sous-jacentes, et un classificateur à une classe pour combiner les caractéristiques apprises dans un indicateur qui représente la déviation du comportement normal du système. Une fois qu'une déviation est identifiée, les caractéristiques sont utilisées pour isoler les composantes du signal les

plus déviantes. Les performances en matière d'isolation des défauts souffrent souvent du fait que les conditions de défaut affectent une grande partie des signaux mesurés, masquant ainsi la source du défaut. Pour y remédier, Chao et al [29] ont proposé une approche hybride combinant des modèles-physique et des algorithmes d'apprentissage profond. Des variables de processus non observées sont déduites à l'aide d'un modèle-physique pour améliorer l'espace d'entrée du modèle d'apprentissage profond. L'espace d'entrée accru qui en résulte gagne en puissance de représentation, ce qui permet une détection et une isolation plus précises des défauts.

**Identification de la cause des défauts** L'identification de la cause des défauts est la partie qui détermine la sévérité, le type et l'attribut du défaut après la détection et l'isolation du défaut. Après l'identification de la cause des défauts, le type et la taille du défaut du système peuvent être obtenus, ce qui fournit des informations détaillées sur le défaut utile pour la partie de réparation du défaut. Plus la précision de l'identification de la cause des défauts est élevée, plus l'estimation de la gravité des défauts par le système de diagnostic est précise, et plus l'évaluation des défauts et la prise de décision sont facilitées.

Afin de résoudre le problème de l'impossibilité d'appliquer des diagrammes de contribution traditionnels aux processus non linéaires, Liu et al [96] ont étendu la méthode classique d'identification de la cause des défauts basée sur la reconstruction à l'analyse en composantes indépendantes du noyau et ont développé la méthode de reconstruction des défauts unidimensionnels aux défauts multidimensionnels. Cette méthode est utilisée pour indiquer l'ampleur des défauts revenant à la normale comme indice d'identification des défauts, ce qui est plus direct que l'indice d'identification des défauts existant. Sun et al [157] ont utilisé un réseau neuronal profond pour trouver la différence de caractéristiques des défauts entre les différents modes de données, puis ont utilisé une fenêtre coulissante pour amplifier le rapport de bruit des défauts et la différence de caractéristiques, ce qui a amélioré efficacement les performances de reconnaissance des défauts du système.

**Réparation des défauts** Une fois qu'un défaut a été correctement diagnostiqué, l'approche optimale pour le résoudre peut ne pas être évidente. En effet, diverses approches de résolution (manuelles ou automatiques) peuvent être envisagées selon la gravité, la répétabilité et la localisation du défaut. Par exemple, dans le cas d'un problème de capteur, une technique de reconstruction de capteur peut être appliquée au processus pour

restaurer les opérations de contrôle [51].

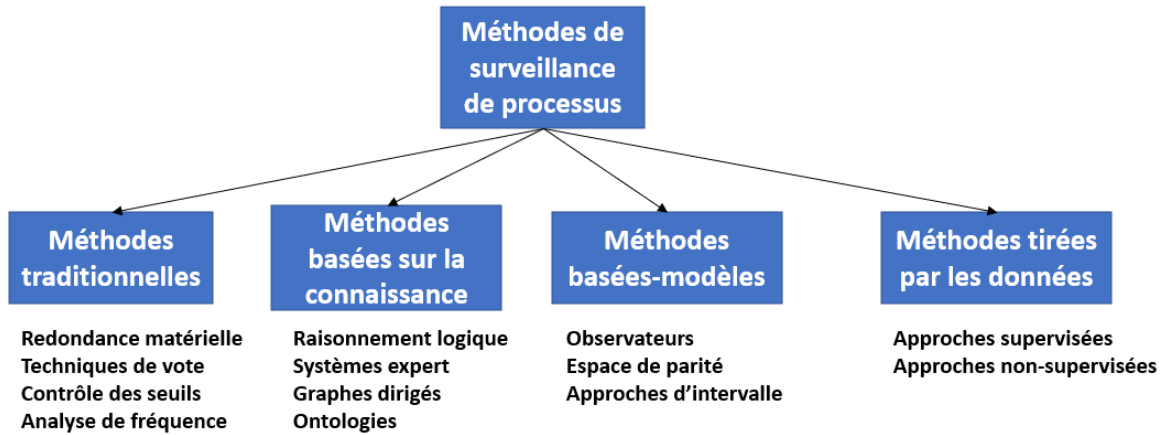


FIGURE 1.3 – Méthodes de surveillance de processus

## 1.2 Méthodes de surveillance process

Dans cette partie, les méthodes de détection de défauts seront détaillées. Ces dernières sont classées en 4 catégories (figure 1.3) : les méthodes traditionnelles, les méthodes basées sur la connaissance, les méthodes basées sur des modèles physique et les celles basées sur les données.

### 1.2.1 Méthodes traditionnelles

Les méthodes traditionnelles se basent sur de simples critères de surveillance facilement applicable en industrie. Nous retrouvons :

**Redondance matérielle [52]** La redondance matérielle est l'approche traditionnelle du diagnostic des défauts qui utilise plusieurs capteurs et actionneurs afin de mesurer une variable d'intérêt particulière. Un inconvénient majeur de la redondance matérielle est l'équipement supplémentaire (capteurs et actionneurs), le poids supplémentaire et le coût de maintenance qui y sont associés. En outre, l'espace supplémentaire requis pour accueillir l'équipement fait de la redondance matérielle une méthode peu populaire pour le diagnostic des défauts.



**Techniques de vote [66]** Les techniques de vote sont souvent utilisées dans les systèmes incorporant un degré élevé de redondance matérielle parallèle. Les techniques de vote sont assez faciles à mettre en œuvre et conviennent surtout au diagnostic de défauts dans les instruments présentant des défauts mécaniques. Pour décrire le fonctionnement d'une technique de vote, considérons trois capteurs identiques mesurant la même variable. Si l'un des trois signaux diffère nettement des deux autres, ce signal est identifié comme défectueux. La différence entre les deux signaux de chaque paire de capteurs dans un groupe redondant indique un défaut.

**Contrôle des seuils [110]** En utilisant le contrôle des seuils, les variables du processus sont mesurées et comparées à des limites connues pour chaque variable. Généralement, la première étape consiste à établir le seuil des variables, puis à les comparer aux valeurs mesurées. Toute mesure ou comparaison entre le seuil connu et la valeur mesurée en dehors de la plage prévue indique la présence d'un défaut. Un exemple simple est le cas d'un détecteur de fumée domestique où la sonnerie se déclenche lorsque la fumée dans la maison a atteint un seuil prédéfini.

**Analyse de fréquence [148]** L'analyse de fréquence des mesures du plan peut être utilisée avec succès dans le diagnostic des défauts des systèmes dynamiques. La plupart des variables de l'installation présentent un spectre de fréquence typique dans des conditions de fonctionnement normales. Toute déviation de ce spectre peut être interprétée comme une anomalie. Certains types de défauts peuvent même avoir leur signature caractéristique dans le spectre, ce qui facilite l'isolation directe des défauts.

## 1.2.2 Méthodes basées sur la connaissance

Avec le développement de l'informatique, le raisonnement logique ainsi que d'autres techniques ont été exploités dans des systèmes d'aide à la décision à grande échelle. En effet, des méthodes basées sur la connaissance, telles que l'analyse causale [112], les systèmes experts [86] et les ontologies [181] ont été proposées durant les deux dernières décennies. Ces techniques sont basées sur des modèles qualitatifs, qui peuvent être obtenus à partir d'une modélisation causale du système, de connaissances d'experts, d'une description détaillée du système ou d'exemples de symptômes de défauts. Les techniques d'analyse causale sont basées sur la modélisation causale des relations défauts-symptômes.

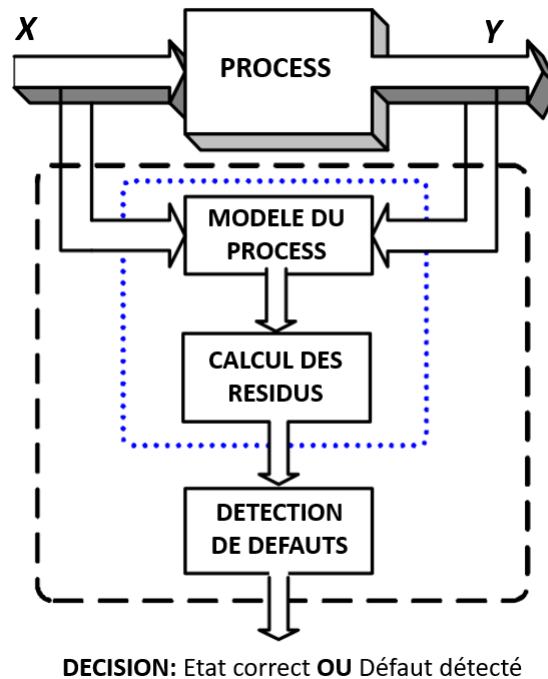


FIGURE 1.4 – Schéma général de la détection des défauts basée sur un modèle physique [98]

Les techniques d'analyse causale, notamment les graphes dirigés signés et l'arbre des symptômes, sont principalement utilisées pour le diagnostic des défauts. Les techniques les plus simples consistent en des arbres de règles logiques du type "SI-symptôme-ET-symptôme-ALORS-conclusion" [128]. Chaque conclusion peut, à son tour, servir de symptôme à la règle suivante, jusqu'à ce que la conclusion finale soit atteinte. Le système peut traiter les informations présentées par le matériel/logiciel de détection ou interagir avec un opérateur humain, lui demandant des informations sur des symptômes particuliers et le guidant tout au long du processus logique. La description des méthodes de diagnostic de défauts basées sur des ontologies est décrite dans le chapitre 3.

### 1.2.3 Méthodes basées-modèle

Les méthodes de surveillance basées sur un modèle reposent sur la comparaison des mesures réelle du processus avec celles d'un modèle physique du processus, qui est généralement dérivé en utilisant une certaine compréhension fondamentale du processus opérant dans des conditions normales. Les résidus, qui sont les différences entre les mesures réelles et les prédictions du modèle, peuvent être utilisés comme un indicateur de l'existence

ou de l'absence de défauts [84](figure 1.4). Lorsque le processus surveillé est dans des conditions de fonctionnement normales, les résidus sont nuls ou proches de zéro en cas d'incertitudes de modélisation et de bruit de mesure. Cependant, lorsqu'un défaut se produit, les résidus s'écartent significativement de zéro, indiquant la présence d'une nouvelle condition qui se distingue de manière significative du mode de fonctionnement normal sans défaut [84]. Les approches de surveillance basées sur des modèles comprennent les méthodes basées sur des observateurs [95], les approches d'espace de parité [122] et les approches d'intervalle [10]. Bien évidemment, l'efficacité de ces méthodes de surveillance basées sur des modèles dépend de la précision des modèles utilisés. Dans la plupart des cas, la modélisation d'un composant d'un système industriel est difficilement réalisable (voir impossible). Des méthodes basées sur les données qui ne requièrent pas de connaissances a priori du système peuvent en revanche être utilisées. Ces méthodes seront détaillées en profondeur dans la prochaine section.

### 1.3 Méthodes basées sur les données

Les méthodes de diagnostic des défauts basées sur les données ne déploient pas un modèle mathématique de la physique de l'usine ou du processus, mais un modèle dérivé des données d'entrée et de sortie connues et mesurées du processus. L'idée fondamentale du diagnostic des défauts basé sur les données est de générer un modèle du processus, qui relie mathématiquement les entrées mesurées aux sorties mesurées, puis d'utiliser ce modèle contre le processus réel pour effectuer la surveillance.

Le principal avantage des approches basées sur les données est que les connaissances physiques relatives au processus ne sont pas nécessaires. Cependant, les données pertinentes de l'historique des opérations process sont supposées mesurées et disponibles. Comme son nom l'indique, l'efficacité de ces approches dépend de la qualité et de la quantité des données du processus. Une grande quantité de données est facilement disponible dans l'industrie des procédés, en raison de l'Industrie 4.0, avec les transformations digitales des entreprises, la multiplication des capteurs sur les machines et des connectivités accrues pour les échanges d'information avec les machines. Cela a stimulé le développement d'approches basées sur les données pour la détection de défauts et la surveillance process. En effet, ces dernières ont été largement utilisées dans de nombreuses applications industrielles [109, 164, 29].

Les méthodes de détection de défauts basées sur les données peuvent être classées en deux catégories principales, à savoir les approches supervisées et les approches non-supervisées.

### 1.3.1 Approches supervisées

L'objectif des méthodes basées sur l'apprentissage supervisé est d'apprendre une fonction (un modèle mathématique) qui fait correspondre une (ou des) donnée(s) d'entrée (mesure de capteur) à une sortie (état d'une machine) sur la base d'exemples de paires d'entrée-sortie. En effet, chaque exemple est une paire constituée d'un objet d'entrée (généralement un vecteur) et d'une valeur de sortie souhaitée (étiquette ou valeur réelle). Un algorithme d'apprentissage supervisé analyse les données d'apprentissage et produit une fonction déduite, qui peut être utilisée pour prédire la valeur de la sortie d'un nouvel exemple à partir de sa donnée d'entrée. Ainsi, ces méthodes requièrent un ensemble d'apprentissage où les données d'entrée sont étiquetées. En particulier, les problèmes d'apprentissage supervisé sont divisés en deux catégories : "classification" et "régression".

Dans un problème de régression, l'algorithme est entraîné pour approximer une correspondance entre les données d'entrée et la variable de sortie prenant des valeurs réelles. Tandis que dans un problème de classification, la variable de sortie est une étiquette/catégorie de sortie. Les problèmes de détection de défauts sont souvent associés à des problèmes de classification [1]. Cette catégorie représente la majorité (79%) de toutes les implémentations dans l'industrie [114]. En effet, de nombreuses techniques (machine à vecteur de support (SVM) [127], réseaux de neurones artificiels (ANN) [4], Réseaux Bayésiens (BN) [20]) se sont progressivement développées au cours des dernières décennies. En revanche, du fait de l'augmentation de la puissance de calcul et des données volumineuses, certaines techniques sont devenues plus populaires ces dernières années. L'apprentissage profond est l'un d'entre elles, qui consiste essentiellement à utiliser des ANN à plusieurs couches cachées.

En effet, l'apprentissage profond est généralement décrit comme une application des réseaux de neurones aux tâches d'apprentissage qui contiennent plus d'une couche cachée. Du fait de leurs généralités dans le sens où ils peuvent approximer une distribution quelconque avec un nombre suffisant de couches, ces techniques ont prouvé leur efficacité dans de nombreux domaines notamment dans la vision par ordinateur [162], la

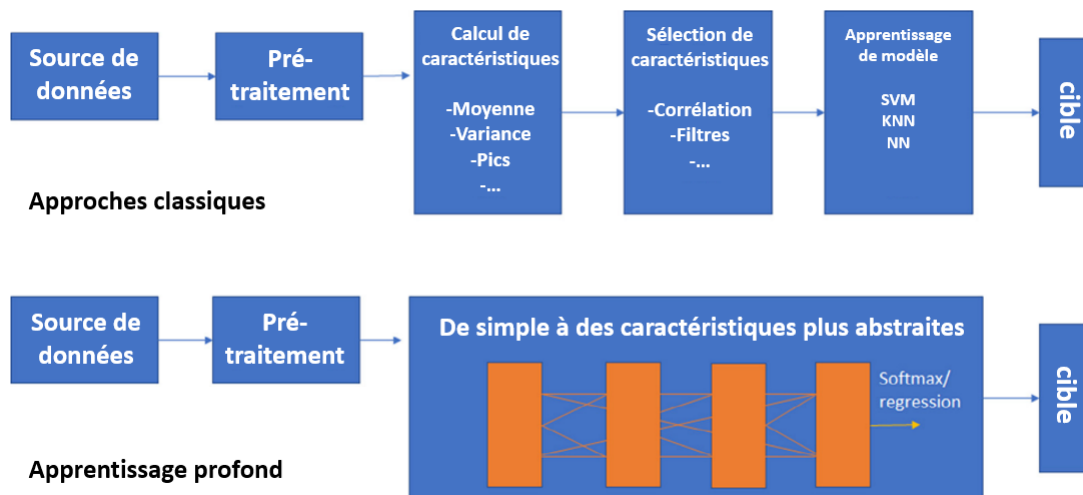


FIGURE 1.5 – Une vue d’ensemble de l’apprentissage conventionnel et de l’apprentissage profond [82]

reconnaissance vocale [44], et la détection de défauts [109]. De plus, à la différence des techniques classiques qui requièrent de la connaissance humaine pour définir un ensemble de caractéristiques pertinentes, l’apprentissage profond peut extraire des caractéristiques significatives directement à partir des données d’entrée en s’appuyant sur ses couches cachées. Cette spécificité rend ses modèles moins dépendants de la connaissance métier et peuvent être utilisés pour implémenter un système de surveillance de bout en bout (de la donnée brute au modèle final, figure 1.5). En revanche, malgré ses avantages, il existe certaines limitations qui peuvent restreindre l’application en milieu industriel.

En effet, la plupart des approches sont spécifiques à une application ou à un équipement et il n’existe pas de méthode claire pour sélectionner, concevoir ou mettre en œuvre une architecture d’apprentissage profond [82]. De plus, de nombreuses architectures d’apprentissage profond sont utilisées et appliquées pour résoudre des problèmes de diagnostic spécifiques. Les chercheurs n’ont pas expliqué, ni documenté, les raisons pour lesquelles ou comment ces architectures ont été sélectionnées [82]. Par ailleurs, l’accent n’est pas mis sur le coût de la mise en œuvre de ces architectures qui sont connus pour leurs exigences en matière de ressources de calculs [58]. En outre, ces méthodes sont considérées comme des boîtes noires dans le sens où il est difficile d’interpréter les résultats de la détection. Enfin, l’intégration de connaissance métier est difficilement réalisable avec ce type d’approche. Récemment, des approches tentant d’intégrer des connaissances dans ces architectures ont été réalisées [166]. Cependant, elles continuent de dépendre fortement

de l'architecture boîte-noire et réduisent la confiance vis-à-vis des industriels quant à une implémentation réelle. Enfin, il convient de souligner que ces modèles exigent une grande quantité de données.

Globalement, bien que les méthodes d'apprentissage supervisé atteignent une grande précision de diagnostic [58], leur application est limitée en raison de l'écart significatif entre les exigences du laborieux étiquetage des données et les données industrielles réelles, car les données historiques des usines manquent généralement de procédures d'étiquetage des données (manque de personnel et/ou de temps) [1]. De plus, la présence de valeurs aberrantes constitue un autre défi dans les jeux de données du monde réel. Contrairement aux classifications traditionnelles, la vérité terrain est souvent indisponible pour la détection des valeurs aberrantes. De ce fait, les techniques d'apprentissage non-supervisés sont alors envisagées.

### 1.3.2 Approches non-supervisés

Les méthodes d'apprentissage non-supervisé n'exigent pas données étiquetées (pas de données de sortie), ce qui les rendent plus attractives dans le cadre de surveillance process. L'objectif de ces techniques est de modéliser la structure ou la distribution sous-jacente des données sans utiliser d'étiquettes explicitement fournies. Les problèmes d'apprentissage non-supervisé sont généralement regroupés en deux catégories : la réduction de la dimensionnalité et le clustering.

Dans un un problème de réduction de la dimensionnalité, l'algorithme découvre la description des caractéristiques latentes communes des données qui interagissent avec la représentation initiale des caractéristiques éparses. Ces représentations latentes serviront comme de nouvelles caractéristiques qui peuvent être exploitées par des modèles supervisés ou non-supervisé. Des techniques comme l'analyse en composantes principales (ACP) [96] ainsi que les auto-encodeurs de l'apprentissage profond [82] sont classées comme techniques de réduction de dimensionnalité.

Dans un problème de clustering, l'algorithme apprend les regroupements inhérents aux données. En effet, l'objectif est de regrouper les observations similaires et de distancier les observations différentes en se basant sur des mesures de similarité. La sortie de ces algorithmes produira des clusters dans les données regroupant des observations similaires dans chaque cluster. Dans le cadre de la détection de défauts, chaque cluster peut décrire

un défaut particulier dans le jeu de donnée [181].

Au cours des deux dernières décennies, beaucoup de travaux de recherche préconisent le déploiement de méthodes non-supervisées pour la détection des défauts [177, 108]. En revanche, plusieurs limitations existent et affectent négativement la confiance quant à l'implémentation de ces méthodes en industrie.

### 1.3.3 Discussion

Malgré leur succès, la fiabilité des méthodes basées sur les données est encore remise en question par les industriels. En effet, les modèles d'apprentissage sont entraînés pour obtenir les meilleures performances sur un jeu de données d'entraînement. Ces derniers peuvent sur-apprendre sur les données, ce qui résulte en une faible généralisation. À l'heure actuelle, il n'existe pas d'approche efficace pour détecter les problèmes de sur-apprentissage sur des données non-étiquetées. Par conséquent, la bonne performance des modèles n'est pas garantie en pratique.

Par ailleurs, le taux d'apparition de défauts en industrie est généralement très faible par rapport au taux des opérations normales, ce qui résultent en des jeux de données très déséquilibrés. Ce problème est connu pour affecter significativement les méthodes de surveillance basées sur les données. Quelques approches ont été proposées [166] mais restent limitées par rapport aux limites énoncées dans la section 1.3.1.

En outre, les approches proposées sont souvent ponctuelles dans le sens où de bons résultats sont obtenus sur un jeu de données initial. En revanche, comme les données arrivent en flux continue en industrie, ces dernières peuvent comprendre des observations de nouveaux défauts, ou une dérive des défauts déjà appris. Ainsi, un modèle de classification classique, qui a été entraîné sur un jeu de données initial, pour un nombre donné de défauts connus, ne pourra pas détecter ces nouveaux comportements. Dans la littérature, il existe trois types d'approches pour détecter de nouveau comportement : le seuillage d'un état correct, ou le rejet à distance avec ou sans considération probabiliste.

Pour la première, seules les observations classées comme bonnes sont considérées pour l'entraînement d'un classifieur à une classe et le réglage d'un seuil de rejet en fonction de la variabilité du process [109]. En revanche, les défauts ne peuvent pas être distingués entre eux et la condition que des données soient correctes durant l'apprentissage n'est pas toujours garantie (bruits, défauts, etc...). La seconde effectue un clustering sur un

jeu de données d'entraînement qui comprend certains défauts, puis compare une nouvelle observation à chaque modèle de défaut en utilisant un rejet basé sur la distance. Cette méthode a été appliquée à la cyber-sécurité [55] ou à la détection de défauts de machines avec les C-Means flous [55] et de défauts de fabrication de circuit imprimé par des réseaux de neurones convolutifs (CNN) [33]. Mais, à la connaissance des auteurs, cela n'a jamais été testé en fabrication mécanique. Cette dernière dépend fortement du seuil de distance, qui est difficile à ajuster, en particulier si la variabilité n'est pas prise en compte [2].

De plus, une fois les nouveaux événements détectés, il est parfois difficile d'intégrer la nouvelle classe détectée dans le modèle initial. En effet, une approche naïve consiste à ré-apprendre un nouveau modèle à partir de toutes les données. En revanche, cette dernière peut nécessiter un temps de calcul considérable si un grand modèle a été utilisé (ex : CNN). De plus, il est nécessaire de s'assurer que les données restent disponibles durant toute la période de surveillance afin de permettre ce ré-entraînement du modèle, ce qui n'est pas toujours le cas en réalité. Enfin, il est probable que, pour des raisons de sécurité, les fournisseurs de données ne veulent pas communiquer leurs données historiques en permanence.

Par ailleurs, les Modèles de Mélange de Gaussiennes (GMM) permettent de détecter de nouveaux événements. De plus, ils permettent d'ajouter facilement les classes nouvellement détectées dans le modèle initial. Pour détecter l'émergence de nouveaux événements, deux approches principales ont été proposées. La première repose sur un critère de rejet qui évalue la distance entre une nouvelle observation et chaque gaussienne. Cette approche a été appliquée dans différents domaines comme la reconnaissance de l'activité humaine [16] ou la dérive de la demande en électricité [117], mais rarement dans le domaine de la fabrication. Une autre approche est basée sur la log-vraisemblance moyenne et a été appliquée pour la détection de la dérive dans le tournage [48], mais pas pour la détection de défauts inconnus. Nous détaillerons ces approches dans le chapitre 2.

En plus de la possibilité de détecter et d'inclure de nouveaux événements, les GMM sont connus pour leur généralité dans le sens où ils peuvent approximer une distribution quelconque avec un nombre suffisant de gaussiennes [64], ce qui les rendent idéales pour la surveillance de procédés industriels. Par ailleurs, les GMM peuvent être appris en non-supervisés, ce qui ne requiert pas d'étiquettes, et ces derniers peuvent être utilisés pour classer de nouvelles observations une fois appris. Enfin, ces modèles sont connus pour être robustes vis-à-vis des jeux de données déséquilibrés, ce qui est le cas en monitoring



de procédé de fabrication. Par conséquent, les GMM sont très adaptés aux contraintes industrielles liées à la surveillance process. De ce fait, nous allons nous baser sur ces modèles dans nos contributions. Nous détaillerons les concepts liés aux GMM dans le prochain chapitre.

## 1.4 Conclusion

Dans ce premier chapitre, nous avons commencé par exposer les notions de base en matière de surveillance process. En effet, les types de défauts ainsi que le processus de diagnostic des défauts ont été présentés. Ensuite, un aperçu global sur les méthodes de surveillance process a été détaillé, avec un accent centré autour des méthodes basées par les données. En outre, nous avons évoqué les limites des approches actuelles, à savoir :

- Les données issues des opérations de fabrication sont non-labélisées et déséquilibrés,
- La capacité du modèle de surveillance de détecter et d'intégrer de nouveaux défauts,
- La surveillance et la réduction du sur-apprentissage des modèles dans le temps,
- La multitude de modèles de surveillance existants et le fait qu'aucun consensus n'existe quant au meilleur modèle à utiliser.

Enfin, nous avons illustré l'intérêt croissant des GMM dans la surveillance process. Le chapitre 2 est consacré aux notions associées à ces modèles.

# MODÈLE À MÉLANGE DE GAUSSIENNES

---

Dans le chapitre 1, nous avons mis en évidence la diversité des méthodes de détection de défauts dans la fabrication industrielle. Au-delà de leurs limites, nous avons vu que les modèles à mélange de gaussiennes (GMM) suscitent un intérêt croissant en raison de leur généralité et simplicité. Dans ce chapitre, les algorithmes d'apprentissage associés à ce formalisme sont présentés. Ensuite, notre étude va se concentrer sur deux aspects appliqués aux GMM : les mécanismes de décisions avec rejet et les techniques de réduction de GMM.

## 2.1 Modèle de Mélange de Gaussiennes

Dans cette section, un aperçu de la modélisation par mélange de gaussiennes est donné, suivant par la présentation de l'algorithme d'apprentissage et des critères de sélection.

### 2.1.1 GMM

Récemment, les modèles à mélange de gaussiennes (GMM) ont été largement utilisés pour le clustering dans de nombreuses applications, telles que l'identification et la vérification des locuteurs [135, 136], la segmentation des images [15, 123] et le suivi des objets [155]. Leur principale utilisation est pour faire de l'apprentissage non-supervisé, c'est à dire découvrir des structures cachées dans les jeux de données.

Une distribution gaussienne, également connue sous le nom de distribution normale, est largement utilisée pour modéliser une distribution des variables continues. Dans le cas d'une variable uni-dimensionnelle  $X$ , la distribution gaussienne peut être écrite sous la forme suivante :

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.1)$$

Où  $\mu, \sigma^2$  sont respectivement la moyenne et la variance de la distribution.

Pour un vecteur  $X$  de dimension  $d$ , la distribution gaussienne multivariée prend la forme :

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (2.2)$$

où  $\mu$  est le vecteur moyenne de dimension  $d$ ,  $\Sigma$  est la matrice de covariance de dimension  $d \times d$ , et  $|\Sigma|$  est le déterminant de  $\Sigma$ .

Supposons un ensemble de données  $\mathcal{D} = \{x_1, \dots, x_N\}$  consistant en  $N$  observations d'une variable aléatoire  $X$  de dimension  $d$ . Le GMM suppose que la variable aléatoire  $X$  est distribuée selon un mélange de  $K$  composantes. Chaque composante est représentée mathématiquement par une gaussienne multivariée. Formellement, la fonction de densité de probabilité de  $X$  peut être écrite comme suit :

$$f(x) = \sum_{k=1}^K \pi_k f_k(x); \quad f_k(x) = \mathcal{N}(x|\mu_k, \Sigma_k) \quad (2.3)$$

Où  $K$  est le nombre total de composantes du mélange. Les paramètres  $\mu_k, \Sigma_k$  et  $\pi_k$  sont respectivement le vecteur moyenne, la matrice de covariance et le poids de la  $k^{\text{ème}}$  distribution normale (i.e,  $f_k$ ). Pour satisfaire la condition de toute distribution de probabilité, le GMM décrit par l'équation 2.3 doit satisfaire l'équation suivante :

$$\sum_{k=1}^K \pi_k = 1, \quad 0 \leq \pi_k \leq 1 \quad (2.4)$$

L'ensemble des paramètres du modèle est représenté par  $\Theta_K = \{\{\pi_k, \mu_k, \Sigma_k\}, \text{ où } k \in \{1, \dots, K\}\}$ . Sans fixer de contraintes supplémentaires sur les  $\Sigma_k$  (i.e matrice pleine), le nombre total de paramètres est :

$$\Psi_K = K + Kd + K \frac{d(d-1)}{2} \quad (2.5)$$

Représentant le nombre de paramètres associés aux poids, aux moyennes, aux covariances respectivement.

Plusieurs raisons expliquent l'intérêt de ce type de distribution [64] :

- de nombreux processus naturels suivent une distribution gaussienne
- les distributions gaussiennes sont relativement simples à manipuler d'un point de vue mathématique
- il est possible d'approximer une distribution de probabilité arbitraire en combinant un nombre suffisant de gaussiennes.

Ainsi, avec un nombre de gaussienne fixé à priori, le GMM a pour but d'approximer au maximum la distribution de points formée par l'ensemble  $\mathcal{D}$ . Pour ce faire, l'objectif est d'apprendre l'ensemble  $\Theta_K$  à partir des observations. L'estimation du maximum de vraisemblance (EMV) est couramment utilisée.

## 2.1.2 Maximum de vraisemblance

L'idée principale de l'EMV est d'estimer l'ensemble des paramètres en maximisant la vraisemblance de l'échantillon utilisé. Formellement, en considérant que le nombre de composantes  $K$  est fixe et en supposant que les données sont indépendamment et identiquement distribuées, la fonction de vraisemblance s'écrit sous la forme suivante :

$$\mathcal{L}(\Theta_K, \mathcal{D}) = p(\mathcal{D}|\Theta_K) = \prod_{i=1}^N p(x_i|\Theta_K) = \prod_{i=1}^N f(x_i) = \prod_{i=1}^N \sum_{k=1}^K \pi_k f_k(x_i) \quad (2.6)$$

D'un point de vue analytique, il est souvent plus facile de maximiser la log-vraisemblance plutôt que la vraisemblance elle-même :

$$l(\Theta_K, \mathcal{D}) = \log \mathcal{L}(\Theta_K, \mathcal{D}) = \sum_{i=1}^N \log(p(x_i|\Theta_K)) = \sum_{i=1}^N \log\left(\sum_{k=1}^K \pi_k f_k(x_i)\right) \quad (2.7)$$

Ainsi, les paramètres optimaux sont ceux qui satisfont à l'équation suivante

$$\frac{\partial(l(\Theta_K))}{\partial(\Theta_K)} = 0 \quad (2.8)$$

en appliquant la formule de Bayes, la formule de  $p(k|x_n)$  est notée comme suit :

$$p(k|x_n) = \frac{\pi_k f_k(x_n)}{\sum_{j=1}^K \pi_j f_j(x_n)} \quad (2.9)$$

Il convient de noter que les probabilités a posteriori doivent satisfaire la contrainte :

$$\sum_{k=1}^K p(k|x_n) = 1; \quad \forall n \in \{1, \dots, N\} \quad (2.10)$$

Ainsi, en dérivant l'équation 2.8 suivant les paramètres  $\mu_k, \Sigma_k, \pi_k$ , les formules suivantes sont obtenues :

$$\mu_k = \frac{\sum_{i=1}^N p(k|x_i) x_i}{\sum_{i=1}^N p(k|x_i)} \quad (2.11)$$

$$\Sigma_k = \frac{\sum_{i=1}^N p(k|x_i) (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^N p(k|x_i)} \quad (2.12)$$

$$\pi_k = \frac{\sum_{i=1}^N p(k|x_i)}{N} \quad (2.13)$$

Il convient de noter que les équations 2.11, 2.12 et 2.13 ne sont pas des solutions à forme fermée pour les paramètres du modèle de mélange. La raison en est que ces équations sont couplées à l'équation 2.9. Plus précisément, les probabilités a posteriori  $p(k|x_n)$  données par l'équation 2.9 dépendent de tous les paramètres du modèle de mélange, alors que tous les résultats de 2.11, 2.12 et 2.13 reposent sur  $p(k|x_n)$ . Par conséquent, la maximisation de la fonction de log-vraisemblance pour un modèle de mélange gaussiennes s'avère être un problème très complexe. Une méthode élégante et puissante pour trouver des solutions de vraisemblance maximale pour les GMM est appelée l'algorithme Espérance-Maximisation ou algorithme EM.

### 2.1.3 L'algorithme d'Espérance-Maximisation (EM)

Introduit par Dempster et al [42], l'algorithme espérance-maximisation (EM) permet d'estimer itérativement les paramètres du maximum de vraisemblance d'une distribution lorsque les données observées  $X$  sont incomplètes. Cette méthode suppose l'existence d'un jeu de données non observées  $Z = \{z_1, \dots, z_N\}$  tel que  $(\mathcal{D}, Z)$  constitue les données complètes. En considérant un GMM avec  $K$  gaussiennes, chaque  $z_i$  est un vecteur binaire à  $K$  dimensions prenant une valeur de 1 sur une dimension et 0 sur les autres. Ainsi, l'interprétation de  $z_i$  se traduit par la composante ayant servi à générer l'observation  $x_i$  (i.e, l'étiquette de  $x_i$ ). En partant de l'équation 2.7, nous pouvons la réécrire comme suit :

$$l(\Theta_K) = \log \mathcal{L}(\Theta_K, \mathcal{D}) = \sum_{i=1}^N \log(p(x_i|\Theta_K)) = \sum_{i=1}^N \log \sum_{z_i} p(x_i, z_i|\Theta_K) \quad (2.14)$$

L'astuce essentielle de l'algorithme EM consiste à maximiser une borne inférieure sur la log-vraisemblance au lieu de la log-vraisemblance.

Pour ce faire, l'algorithme exploite l'inégalité de Jensen qui stipule si  $f$  est une fonction convexe,  $(x_1, x_2, \dots, x_n)$  un  $n$ -uplet de réels appartenant à l'intervalle de définition de  $f$  et  $(\lambda_1, \dots, \lambda_n)$  un  $n$ -uplet de réels positifs tels que  $\sum_{i=1}^n \lambda_i = 1$  alors :

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i) \quad (2.15)$$

L'inégalité 2.15 est inversée dans le cas où la fonction  $f$  est concave.

Ainsi, en exploitant la concavité de la fonction  $\log$  et en définissant une distribution arbitraire  $q(z_j)$  définie sur les variables latentes ( $\sum_{z_j} q(z_j) = 1, q(z_j) \geq 0$ ), la décomposition suivante tient sur la base de l'inégalité de Jensen 2.15 :

$$\begin{aligned} l(\Theta_K) &= \log p(X|\Theta_K) = \sum_{i=1}^N \log \sum_{z_i} p(x_i, z_i|\Theta_K) \\ &= \sum_{i=1}^N \log \sum_{z_i} q(z_i) \frac{p(x_i, z_i|\Theta_K)}{q(z_i)} \\ &\geq \sum_{i=1}^N \sum_{z_i} q(z_i) \log \frac{p(x_i, z_i|\Theta_K)}{q(z_i)} \equiv \mathcal{L}(q, \Theta_K) \end{aligned} \quad (2.16)$$

Il est évident que  $\mathcal{L}(q, \Theta_K)$  est une borne inférieure de  $\log p(X|\Theta_K)$ . On note que  $\mathcal{L}(q, \Theta_K)$  est une fonction de la distribution  $q(z)$  et une fonction des paramètres  $\Theta_K$ . Il est intéressant d'étudier attentivement qu'il sera plus facile de maximiser cette borne inférieure sur la log-vraisemblance que la log-vraisemblance réelle.

En supposant  $q(z_i) = p(z_i|x_i, \Theta_K)$ , on obtient :

$$\begin{aligned} \mathcal{L}(q, \Theta_K) &= \sum_{i=1}^N \sum_{z_i} p(z_i|x_i, \Theta_K) \log \frac{p(x_i, z_i|\Theta_K)}{p(z_i|x_i, \Theta_K)} \\ &= \sum_{i=1}^N \sum_{z_i} p(z_i|x_i, \Theta_K) \log p(x_i|\Theta_K) \\ &= \sum_{i=1}^N \log p(x_i|\Theta_K) = l(\Theta_K) \end{aligned} \quad (2.17)$$

Par conséquent, avec ce choix de  $q$ , la borne inférieure  $\mathcal{L}(q, \Theta_K)$  sera égale à la log-vraisemblance, ce qui indique que la borne  $\mathcal{L}(q, \Theta_K)$  est serrée (atteint l'égalité) pour un  $\Theta_K$  fixé. De plus, puisque  $\mathcal{L}(q, \Theta_K) \leq l(\Theta_K)$ , ce choix de  $q$  maximise  $\mathcal{L}(q, \Theta_K)$  pour l'ensemble des paramètres  $\Theta_K$  fixé.

Ainsi, à partir des paramètres initiaux  $\Theta_K^0$  fixés a priori, l'algorithme EM opère, à chaque itération  $t$ , deux étapes successives :

- l'étape d'espérance (E), où la borne inférieure  $\mathcal{L}(q, \Theta_K)$  est maximisée par rapport à  $q(z_i)$  en gardant  $\Theta_K^{(t)}$  fixé, ce qui peut être exprimé sous la forme :

$$q^{(t)}(z_i) = \arg \max_q \mathcal{L}(q, \Theta_K^{(t)}) = p(z_i|x_i, \Theta_K^{(t)}); \quad \forall i \in \{1, \dots, N\} \quad (2.18)$$

Cela revient à calculer les probabilités a posteriori données par l'équation 2.9

- l'étape de maximisation (M), où sont estimés les paramètres  $\Theta_K^{(t)}$  qui maximise  $\mathcal{L}(q^{(t)}, \Theta_K)$  en gardant  $q^{(t)}(z_j)$  fixés :

$$\Theta_K^{(t+1)} = \arg \max_{\Theta} \mathcal{L}(q^{(t)}, \Theta_K) \quad (2.19)$$

La solution du problème 2.19 revient à calculer les équations 2.11, 2.12, 2.13.

Comme le montrent Dempster et al [42], chaque itération de l'algorithme EM augmente la log-vraisemblance jusqu'à ce que celle-ci converge vers un maximum local.

### 2.1.4 Limitations de l'algorithme EM

La popularité de l'algorithme EM est due à son implémentation simple et la garantie de l'augmentation de log-vraisemblance de l'ensemble d'apprentissage pendant l'optimi-

sation. Cependant, l'algorithme EM présente plusieurs limites bien connues.

L'algorithme EM converge vers un maximum local et sa solution dépend fortement de l'initialisation, ce qui produit par conséquent des estimations de maximum de vraisemblance sous-optimales. Pour y remédier, une stratégie simple consiste à utiliser plusieurs points de départ de façon aléatoire et à choisir l'estimation finale avec la probabilité la plus élevée [105]. Une autre solution consiste à initialiser l'optimisation par des algorithmes de clustering [105]. En outre, de nombreuses adaptations et extensions de l'algorithme EM ont été proposées afin de résoudre les problèmes de convergence vers un optimum local et d'initialisation, comme les versions de recuit déterministe (DA) de l'algorithme EM [160], l'algorithme EM à séparation et fusion [161], l'algorithme EM par composant [25] et les algorithmes EM basés sur la génétique [103].

De plus, l'algorithme EM converge lentement dans certaines situations, ce qui a fait l'objet d'une grande attention récemment. De nombreux algorithmes visant à accélérer la convergence de EM tout en préservant sa simplicité ont été proposés [105, 25]. De plus, l'algorithme EM suppose que le nombre de composantes pour modéliser les distributions est connu. Dans de nombreuses applications pratiques, le nombre optimal de composantes est inconnu et doit être déterminé. Dans de tels cas, un ensemble de modèles candidats est établi en appliquant l'algorithme pour un nombre différent de composantes, et le meilleur modèle est sélectionné selon un critère de sélection de modèle comme discuté dans la section 2.1.5.

Enfin, l'algorithme EM converge vers des optimums locaux souvent très loin de l'optimum global dans des données à très haute dimension. Une technique élégante permettant non seulement d'estimer automatiquement le nombre de composantes et de réduire fortement la dépendance quant à l'initialisation [27] a été proposée. Enfin, l'amélioration de la qualité de l'estimation à travers par des solutions parcimonieuses en passant par une sélection de variables ont été proposées [28, 76].

### 2.1.5 Critères de sélection de modèle

L'algorithme EM suppose un nombre de composantes  $K$  fixé a priori. En pratique, il n'est pas évident de connaître précisément le nombre de composantes dans une distribution de données. En effet, comme pour le problème de sélection de modèles, un mélange avec trop de composantes risque de sur-apprendre la distribution de données, tandis qu'un



mélange avec trop peu de composantes risque de ne pas être assez flexible pour se rapprocher du véritable modèle sous-jacent. Plus encore, la fonction maximum de vraisemblance  $\log p(X|\Theta_{MV})$  est une fonction non-décroissante en fonction du nombre de composantes  $K$ . Ainsi, le critère de l'EMV est inutile pour estimer le nombre de composantes, et certaines méthodes avancées de sélection de modèles ont été proposées.

Il existe deux principaux types de méthodes de sélection de modèles : les méthodes déterministes et les méthodes stochastiques. Les méthodes déterministes commencent généralement par l'obtention d'un ensemble de modèles candidats pour une gamme de valeurs de  $K$  (de  $K_{min}$  à  $K_{max}$ ), que l'on suppose contenir le  $K$  optimal. Le nombre de composants est ensuite choisi en fonction de :

$$K^* = \arg \min_K \{C(\Theta_K, K); \quad K = K_{min}, \dots, K_{max}\} \quad (2.20)$$

où  $\Theta_k$  est une estimation des paramètres du mélange avec  $k$  composantes, et  $C(\Theta_K, K)$  est un critère de sélection du modèle. Un critère très courant peut être exprimé sous la forme :

$$C(\Theta_k, K) = -\log p(X|\Theta_K) + \mathcal{P}(K) \quad (2.21)$$

où  $\mathcal{P}(K)$  est une fonction croissante qui pénalise de fortes valeurs de  $K$ . Par exemple, un choix typique de  $\mathcal{P}(K)$  donne le critère d'information bayésien (BIC) [59] est :

$$\mathcal{P}(K) = -\log p(X|\Theta_K) + \Psi_K \log(N) \quad (2.22)$$

où  $\Psi_K$  est le nombre de paramètres du GMM à  $K$  composantes(cf. équation 2.5).

De nombreux exemples de ces critères ont été utilisés pour les modèles de mélange, notamment les critères d'approximation bayésiens, tels que le critère empirique de Laplace (LEC) [105] et les approches de la théorie de l'information [9], telles que la longueur minimale de description (MDL) de Rissanen [139], le critère de longueur minimale de message (MML) [119] et le critère d'information d'Akaike (AIC) [18]. En outre, plusieurs critères de section de modèle ont été proposés sur la base de la vraisemblance de classification, tels que le critère d'entropie normalisée (NEC) [26] et le critère de vraisemblance de classification intégrée (ICL) [12]. Pour les modèles de mélange fini, une revue complète des approches de sélection de modèles est donnée dans [105].

Outre les méthodes déterministes, les méthodes stochastiques basées sur la chaîne de Markov Monte Carlo (MCMC) peuvent être utilisées pour l'inférence de modèles, soit en mettant en œuvre des critères de sélection de modèles [141], soit en modélisant de manière entièrement bayésienne et en échantillonnant à partir de la distribution a posteriori complète avec  $K$  considéré comme inconnu [138]. De plus, des approches de validation croisée [152] ont également été utilisées pour estimer le nombre de composantes du mélange.

### 2.1.6 Lien entre gaussiennes et clusters

Ainsi, les  $K^*$  gaussiennes apprises peuvent décrire des sous-structures "indépendantes" dans les données initiales. Ces dernières sont souvent appelées groupes ou "clusters". En revanche, il est possible qu'un cluster soit décrit par plusieurs gaussiennes comme le montre la figure 2.1 [13].

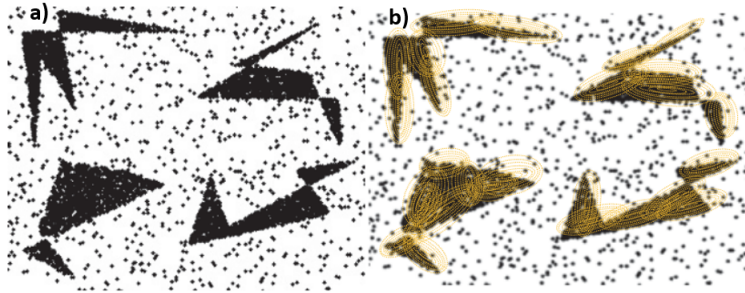


FIGURE 2.1 – Lien entre gaussiennes et clusters : a) distribution réelle, b) approximation de la distribution par des gaussiennes (ellipses jaunes) [13]

Une fois le nombre optimal de gaussiennes  $K^*$  estimé, chaque observation  $x_i$  est affectée par la règle de Bayes à la composante  $C_i$  qui maximise la probabilité a posteriori obtenue par la règle suivante :

$$C_i = \arg \max_k p(k|x_i) = \arg \max_k \pi_k f_k(x_i); \quad \forall i \in \{1, \dots, N\} \quad (2.23)$$

De plus, le GMM appris pourra être utilisé pour classifier de nouvelles observations qui n'ont pas servi à apprendre le modèle. La règle de classification est toujours la même. Elle est donnée par l'équation 2.23. Par exemple, à l'issue de la classification d'une observation

$x$  après un apprentissage sur  $N$  observations initiales, nous pouvons mettre à jour les poids de gaussiennes avec les équations récursives suivantes :

$$\begin{aligned}\pi_k^{N+1} &= \frac{N}{N+1}\pi_k^N + \frac{p(k|x)}{N+1} \quad \forall k \\ N &= N+1\end{aligned}\tag{2.24}$$

Ces équations permettent donc d'augmenter le poids de la gaussienne associée à la classe de  $x$  au détriment des autres gaussiennes. Ce processus permet à terme de réduire les poids des gaussiennes inactives (i.e. obsolètes) et de placer plus d'importance sur les gaussiennes actives.

En revanche, deux cas de figures peuvent se présenter pour une classification de l'observation  $x$  :

- Il est possible aussi que  $x$  soit très éloignée des gaussiennes apprises. Et donc, même si elle a été affectée à la gaussienne la plus proche relativement avec l'équation 2.23, cette affectation peut ne pas être correcte,
- Il est possible que  $x$  soit à la frontière de plusieurs gaussiennes simultanément, et donc, il peut y avoir une ambiguïté à estimer la vraie classe de  $x$ .

Suivant les deux cas, il vaut mieux différer la décision et affecter  $x$  à une classe "d'attente" dite de rejet. On dit alors qu'on a procédé à un rejet de  $x$ , en utilisant une règle de décision avec rejet. Ce formalisme de rejet sera détaillé dans ce qui suit.

## 2.2 Décisions avec rejet

Les mécanismes de décisions avec rejet sont des techniques qui ont suscité un grand intérêt récemment. Leurs principales utilisations résident dans la détection de nouvelles classes qui n'ont pas servi à l'apprentissage du modèle de classification initial. De plus, une autre utilisation consiste à détecter une ambiguïté de classification. Cette ambiguïté traduit par le fait qu'une observation soit à la frontière de plusieurs classes distinctes. La visualisation de ces deux cas d'utilisation est illustrée par la figure 2.2.

Les deux types de rejets appliqués aux GMM seront présentés dans ce qui suit.

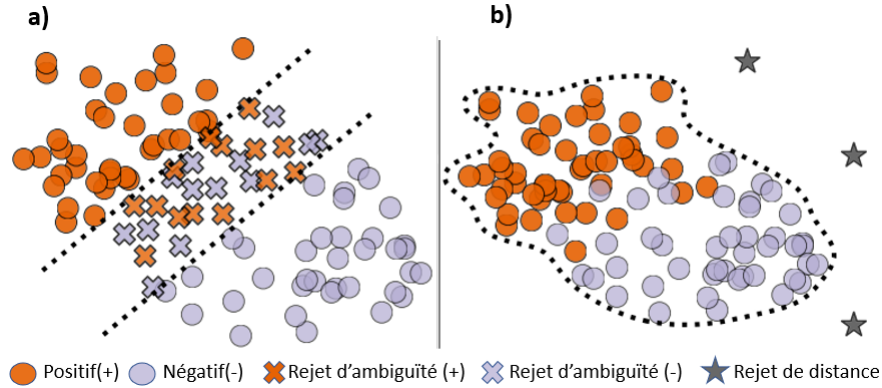


FIGURE 2.2 – Illustration des deux types de rejets (cas d'un modèle linéaire) : a) Rejet d'ambiguïté, b) Rejet de distance [74]

### 2.2.1 Rejet de distance

Pour une nouvelle observation  $x$ , le rejet de distance a pour but de vérifier si elle est assez proche des classes identifiées dans l'apprentissage, afin d'être classée. Dans le cadre d'un GMM à  $K$  composantes, plusieurs méthodes ont été proposées :

**Basées sur la vraisemblance** Engel et al [53] ont proposé une méthode de rejet basée sur la vraisemblance locale de la nouvelle observation  $x$  (i.e,  $f_k(x)$ ). Pour ce faire, ils commencent par fixer un seuil local sur chaque composante, puis à chaque nouvelle observation  $x$ , ils comparent les vraisemblances locales de la nouvelle observation aux seuils locaux. La nouvelle observation est rejetée si seulement si toutes ses vraisemblances locales sont inférieures à leurs seuils locaux. Formellement, l'observation  $x$  est rejetée si et seulement si :

$$f_k(x) < \frac{\tau}{(2\pi)^{d/2} |\Sigma_k|^{1/2}}; \quad \forall k \in \{1, \dots, K\} \quad (2.25)$$

Notons que le seuil local pour chaque composante est juste une fraction du maximum de sa vraisemblance déterminée par  $\tau$ , qui correspond au risque de première espèce, qui est couramment utilisé en Génie Industriel [87]. Engel et al [53] ont recommandé que les valeurs de  $\tau$  soient comprises dans l'intervalle [1%, ..., 10%]. Cette méthode de rejet a été utilisée pour suivre les trajectoires d'un robot [53, 72], suivre l'évolution des prix de l'électricité [118] et de détecter des dérives virtuelles et réels [117] sur plusieurs jeux de données réelles [134].

Une autre méthode basée sur la vraisemblance a été proposée par Diaz-Rozo et al [47]. Cette dernière consiste à comparer la vraisemblance de la nouvelle observation  $x$  à la vraisemblance moyenne du jeux de données d'apprentissage  $\overline{l(\Theta_K)}$  tel que :

$$\overline{l(\Theta_K)} = \frac{1}{N}l(\Theta_K) = \frac{1}{N} \log \mathcal{L}(\Theta_K, X) = \frac{1}{N} \sum_{i=1}^N \log(p(x_i|\Theta_K)) \quad (2.26)$$

Une faible valeur de  $p(x|\Theta_K)$  par rapport à  $\overline{l(\Theta_K)}$  indique que la nouvelle observation est mal représentée par le GMM et est sera rejetée. Ce méthode a été utilisée pour la détection de dérives [47] pour la maintenance en usinage [150] ainsi que pour la détection de dérives dans l'usinage des boites de vitesses [48].

**Basées sur la distance** La deuxième méthode consiste à utiliser la distance de la nouvelle observation  $x$  à des critères de rejets bien définis. Deux approches existent dans la littérature.

La première consiste à calculer la distance de Mahalanobis  $d(x, k)$  pour chaque composante  $k$  comme suit :

$$d(x, k) = \sqrt{(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)} \quad (2.27)$$

Puis, il suffit de comparer cette distance à un seuil fixé  $S$ . Formellement, une observation est rejetée si et seulement si :

$$d(x, k) > S; \quad \forall k \in \{1, \dots, K\} \quad (2.28)$$

Une première proposition de ce critère de rejet (équation 2.28) a été faite par Bouchachia et al [16] pour la reconnaissance de l'activité humaine dans un contexte de maisons intelligentes. Le seuil a été appelé seuil de proximité  $T_\Sigma$  et a été fixé à plusieurs valeurs arbitraires. Selon Bouchachia et al [16], de faibles valeurs de ce seuil donnent de meilleurs résultats. En revanche, il n'y a pas eu d'indication claire pour choisir la valeur du seuil.

La deuxième proposition consiste à comparer le carré de la distance  $d(x, k)$  à  $\chi_{d,1-\beta}^2$  qui représente le  $(1-\beta)$ -percentile d'une distribution  $\chi^2$  avec  $d$  degrés de liberté (où  $d$  est la dimension du vecteur  $x$  et  $\beta$  est un paramètre fixé par l'utilisateur, exemple 0.1). Pinto et al [125] ont été les premiers à proposer cette méthode qui a été appliquée pour apprendre

des GMM performant de façon incrémentale et très rapide. La même approche a été ensuite utilisée dans l'approximation des fonctions et de contrôle dans l'apprentissage par renforcement [124]. Enfin, Chamby et al [80] l'ont utilisé pour apprendre un GMM de façon incrémentale dans un contexte de flux de données non-stationnaires.

Enfin, une deuxième méthode de distance compare la distance euclidienne du vecteur  $x$  par rapport vecteurs moyennes de chaque composante du GMM. Formellement, pour une composante  $k$ , elle est mesurée comme suit :

$$d_2(x, k) = \sqrt{(x - \mu_k)^T (x - \mu_k)} \quad (2.29)$$

Ensuite, comme pour la méthode précédente, une observation  $x$  est rejetée si seulement si :

$$d_2(x, k) > S; \quad \forall k \in \{1, \dots, K\} \quad (2.30)$$

Tel que  $S$  est un seuil fixé par l'utilisateur.

Cette méthode de rejet a été appliquée pour apprendre une arboréscence de GMM de façon incrémentale dans un contexte de flux de données continue [163]. Le seuil de proximité a été fixé de façon arbitraire sur une plage de valeurs.

## 2.2.2 Rejet d'ambiguïté

Concernant le rejet d'ambiguïté, nous n'avons pas trouvé de travaux qui appliquent le rejet d'ambiguïté dans le cadre des GMM. Par conséquent, nous allons présenter dans ce qui suit les principales techniques qui sont utilisées dans d'autres modèles. Nous nous baserons ensuite sur une de ces techniques dans nos contributions.

**Règle générale** Selon Leray et al [94], Il est possible d'utiliser la règle de décision (équation 2.23) avec rejet d'ambiguïté dans un classifieur de type GMM. Pour ce faire, la règle de décision suivante est utilisée :

$$C_i = \begin{cases} \arg \max_k p(k|x_i) & \text{si } m(x_i) \geq \Theta_a \\ C_0 & \text{sinon} \end{cases} \quad (2.31)$$

où  $C_0$  est la classe de rejet d'ambiguïté,  $\Theta_a$  est un seuil arbitraire, et  $m(x_i)$  est une mesure de confiance heuristique vérifiant les hypothèses suivantes :

- $m(x) \in [0, \dots, 1]$
- $m(x) = 0$  lorsqu'il faut toujours rejeter  $x$  (aucune confiance dans cet exemple)
- $m(x) = 1$  s'il faut toujours accepter la décision concernant  $x$  (confiance absolue dans l'exemple).

Plusieurs mesures de confiances peuvent être proposées. Dans ce qui suit, quelques mesures sont présentées.

**Mesures de confiances** La mesure de confiance la plus couramment utilisée (notons la  $m_1$ ) est celle issue de la règle de décision bayésienne (équation 2.23) :

$$m_1(x) = P_{max} = \max_k p(k|x) \quad (2.32)$$

La mesure  $m_1$  prend uniquement en compte la plus grande probabilité a posteriori, mais ne prend pas en compte les probabilités a posteriori des autres classes. En effet, une mesure de confiance peut être meilleure qu'une autre selon les valeurs des probabilités non majoritaires. Montrons cela par l'exemple suivant (cas d'une classification à  $K$  classes) :

$$m(x, P_1) > m(x, P_2) \text{ pour } \{P_1(k|x)\} = \{P_{1max}, \frac{1-P_{1max}}{K-1}, \dots, \frac{1-P_{1max}}{K-1}\} \text{ et } \{P_2(k|x)\} = \{P_{2max}, 1 - P_{2max}, 0, \dots, 0\}.$$

Cette dernière contrainte indique que nous avons d'avantage confiance dans un modèle qui minimise les valeurs des probabilités des classes "perdantes".

Ainsi, il est possible de prendre en compte la probabilité a posteriori des autres classes en utilisant une des mesures suivantes :

$$m_2(x) = 1 - \frac{1}{K^*} H(p(k|x)) \quad (2.33)$$

où  $H(p(k|x)) = -\sum_k p(k|x) \log p(k|x)$  est l'entropie des probabilités a posteriori.

$$m_3(x) = P_{max} - P_{2max} \quad (2.34)$$

où  $P_{2max}$  est le deuxième plus grand élément de  $\{p(k|x)\}$ .

Il est possible d'inventer bien d'autres heuristiques de cette sorte, basées sur d'autres types de distance (distance de Kullback-Liebler, mesures de séparabilité [46]). En pratique, comme montré dans Leray et al [94], la plupart de ces mesures sont équivalentes.

### 2.2.3 Conclusion

Dans cette partie, les différents mécanismes de décisions avec rejet de distances appliqués aux GMM ont été présentés. Ensuite, les mécanismes de décisions avec rejet d'ambiguïté dans un cadre général ont été illustrés. Il convient de souligner que ces mécanismes n'ont pas été appliqués à la détection de défaut dans la surveillance de processus de fabrication.

Ces décisions sont très utiles dans le cadre de flux de données continues. Ainsi, à travers ces décisions, nous pouvons penser que d'autres gaussiennes pourraient être apprises sur les données rejetées puis intégrées au GMM initial. Néanmoins, l'ajout continu de nouvelles gaussiennes pourrait conduire à un sur-apprentissage et saturer le GMM global. De plus, plusieurs gaussiennes peuvent devenir redondantes ou obsolètes. Enfin, des gaussiennes pourraient être fusionnées si elles sont très proches, ce qui permettrait de réduire le sur-apprentissage.

## 2.3 Réduction de modèles

Les contraintes énoncées précédemment nécessiteraient donc de simplifier le modèle GMM au cours du temps. Il se trouve que ce problème a été étudié dans la littérature sous le nom de "Gaussian Mixture Reduction (GMR)".

### 2.3.1 Formalisation du problème

Considérant un GMM initial dont la fonction densité est  $f(x|\Theta_{K_a})$ , le problème de réduction du modèle de mélange de gaussiennes (GMR) revient à trouver un GMM réduit dont la fonction densité est  $g(x|\Theta_{K_b})$  et comprenant moins de composantes (i.e  $K_b < K_a$ ), tel que la dissimilarité entre le GMM initial et le GMM réduit soit minimale. Le problème est formellement décrit par :



$$\Theta_{K_b}^* = \arg \min_{\Theta_{K_b}} D(f(x|\Theta_{K_a})||g(x|\Theta_{K_b})) \quad (2.35)$$

Il s'avère ainsi que le GMR soit un problème d'optimisation non-convexe n'admettant pas de solutions sous formules closes. De plus, il existe plusieurs minimums locaux. En outre, le problème 2.35 dépend aussi de la mesure de dissimilarité qui peut ne pas admettre de formule close pour des GMMs. En effet, l'évaluation de la plupart des mesures D nécessite une intégration volumique sur l'ensemble du domaine  $\mathbb{R}^d$ . Enfin, la forme du GMM réduit peut différer selon la mesure D utilisée.

### 2.3.2 Mesures de dissimilarité

Dans cette partie, la présentation de quelques mesures D les plus utilisées dans les algorithmes de GMR existants est fournie.

**Divergence de Kullback-Leibler** Étant données deux distributions de probabilités continues,  $p$  et  $q$  sur  $\mathbb{R}^d$ , la divergence de Kullback-Leibler (KLD) de  $q$  par rapport à  $p$  (également appelée entropie relative différentielle) est définie comme suit [88] :

$$D_{KL}(p||q) = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)} dx \quad (2.36)$$

La KLD mesure la perte d'information lorsque  $p$  est approximée par  $q$ . Il est important de noter que la KLD n'est pas symétrique (i.e,  $D_{KL}(p||q) \neq D_{KL}(q||p)$ ) et qu'elle ne respecte pas l'inégalité triangulaire. De ce fait, la KLD n'est pas considérée comme une distance.

Dans le cas de deux gaussiennes  $\mathcal{N}(\cdot|\mu_i, \Sigma_i), \mathcal{N}(\cdot|\mu_j, \Sigma_j)$  définie dans un espace de dimension  $d$ , la KLD admet la formule explicite suivante :

$$D_{KL}(N_i||N_j) = \frac{1}{2}(tr(\Sigma_j^{-1}\Sigma_i) + \log \frac{|\Sigma_j|}{|\Sigma_i|} + (\mu_j - \mu_i)^T \Sigma_j^{-1}(\mu_j - \mu_i) - d). \quad (2.37)$$

Appliquée aux GMR, le premier argument du KLD est le GMM original, tandis que le second argument est le GMM réduit :

$$D_{KL}(f(\cdot|\Theta_{K_a})||g(\cdot|\Theta_{K_b})) = \int_{\mathbb{R}^d} f(x|\Theta_{K_a}) \log \frac{f(x|\Theta_{K_a})}{g(\cdot|\Theta_{K_b})} dx \quad (2.38)$$

Le principal problème avec l'équation 2.38 est qu'il n'a pas de formule close, de même pour son gradient par rapport à l'ensemble des paramètres  $\Theta_{K_b}$ , utile dans l'optimisation numérique de l'équation 2.35. Une façon d'approximer l'équation 2.38 est d'utiliser une intégration de Monte-Carlo sur un grand sous-ensemble du domaine  $\mathbb{R}^d$ , une procédure exigeante en termes de calcul, en particulier lorsque le nombre de dimension  $d$  est grand. Une revue complète des d'approximation de la KLD pour les GMM a été proposée par S.Cui et al [35].

Dans le cas du GMR, la KLD a été utilisée dans la navigation sur un terrain référencé [143, 5], ainsi que dans l'estimation des états d'une chaîne de Markov dans un problème de suivi de cible [54]; mais pas en production industrielle, à notre connaissance.

**Erreur quadratique intégrale (ISE)** Une autre mesure D largement utilisée entre deux distributions  $p$  et  $q$  est l'erreur carrée intégrale (ISE), également appelée différence carrée intégrale [147], définie comme le carré de l'erreur  $L_2$  :

$$D_{ISE}(p||q) = \int_{\mathbb{R}^d} (p(x) - q(x))^2 dx = ||p - q||_2^2 \quad (2.39)$$

L'ISE est symétrique mais ne satisfait pas l'inégalité triangulaire, et donc n'est pas considérée comme une distance. Par contre, la racine carrée de cette dernière en est une.

En définissant les vecteurs poids  $\pi^a, \pi^b$  comme  $[\pi_1^a, \dots, \pi_{K_a}^a]^T, [\pi_1^b, \dots, \pi_{K_b}^b]^T$ , l'ISE entre 2 GMM est définie comme suit [34] :

$$\begin{aligned} D_{ISE}(f(\cdot|\Theta_{K_a})||g(\cdot|\Theta_{K_b})) &= \int_{\mathbb{R}^d} (f(x|\Theta_{K_a}) - g(x|\Theta_{K_b}))^2 dx \\ &= \int_{\mathbb{R}^d} (f(x|\Theta_{K_a}))^2 dx - 2 \int_{\mathbb{R}^d} f(x|\Theta_{K_a})g(x|\Theta_{K_b})dx + \int_{\mathbb{R}^d} (g(x|\Theta_{K_b}))^2 dx \\ &= J^{aa} - 2J^{ab} + J^{bb} \end{aligned} \quad (2.40)$$

tel que :

$$\begin{aligned} J^{aa} &= (\pi^a)^T H^{aa} \pi^a \\ J^{bb} &= (\pi^b)^T H^{bb} \pi^b \\ J^{ab} &= (\pi^a)^T H^{ab} \pi^b \end{aligned} \quad (2.41)$$

Et les composantes de  $H^{aa}$ ,  $H^{bb}$ ,  $H^{ab}$  sont :

$$\begin{aligned} (H^{aa})_{ij} &= \mathcal{N}(\mu_i^a | \mu_j^a, \Sigma_i^a + \Sigma_j^a); \quad \forall i, j \in \{1, \dots, K_a\} \\ (H^{bb})_{kl} &= \mathcal{N}(\mu_k^b | \mu_l^b, \Sigma_k^b + \Sigma_l^b); \quad \forall k, l \in \{1, \dots, K_b\} \\ (H^{ab})_{ik} &= \mathcal{N}(\mu_i^a | \mu_k^a, \Sigma_i^a + \Sigma_k^a) \end{aligned} \quad (2.42)$$

Ainsi, en s'appuyant sur [2.41, 2.42], l'ISE entre les deux GMM est alors facile à calculer et admet alors une formule close. De plus, comme montré dans [171], le calcul du gradient en fonction des paramètres dans  $\Theta_{K_b}$  admet une formule close, ce qui est utile lorsque la solution par approche numérique de l'équation 2.35 est recherchée.

Plusieurs travaux ayant utilisé l'ISE ont été réalisés dans le cadre du GMR. Par exemple, le GMR basé sur l'ISE a été utilisé pour le suivi d'une cible unique [170], le suivi de plusieurs cibles [171], ainsi que le suivi de plusieurs cibles d'un filtre Bayésien distribué [174].

**Erreur quadratique intégrale normalisée (NISE)** Une mesure de dissimilarité étroitement liée à l'ISE est sa version normalisée, appelée NISE, définie comme suit :

$$D_{NISE}(p||q) = \frac{\int_{\mathbb{R}^d} (p(x) - q(x))^2 dx}{\int_{\mathbb{R}^d} (p^2(x) + q^2(x)) dx} = \frac{\|p - q\|_2^2}{\|p\|_2^2 + \|q\|_2^2} \in [0, 1] \quad (2.43)$$

Lorsqu'elle est appliquée aux GMM  $f$  et  $g$ , en considérant l'expression 2.40 pour l'ISE, on obtient :

$$D_{NISE}(f(\cdot|\Theta_{K_a})||g(\cdot|\Theta_{K_b})) = \frac{J^{aa} - 2J^{ab} + J^{bb}}{J^{aa} + J^{bb}} = 1 - \frac{2J^{ab}}{J^{aa} + J^{bb}} \in [0, 1] \quad (2.44)$$

Comme pour l'ISE, le NISE est symétrique mais ne satisfait pas l'inégalité triangulaire. La propriété du NISE à avoir des valeurs bornées entre 0 et 1 le rend attractif, dans de nombreux contextes, comme dans le traçage d'une cible, comme mesure pour évaluer la précision d'une approximation [34, 146, 175]. Comme pour l'ISE, le gradient du NISE par rapport à  $\Theta_{K_b}$  admet une formule close, car les expressions des équations 2.44, 2.40 utilisent les mêmes estimations  $J^{aa}$ ,  $J^{bb}$ ,  $J^{ab}$ .

**D'autres mesures de dissimilarité** Plusieurs autres mesures de dissimilarité ont été proposées dans la littérature pour traiter le problème du GMR. Nous retrouvons parmi

elles les distances variationnelles de Bhattacharyya et de Kolmogorov [170, 171], la distance de Cauchy-Schwarz [81], la distance de Wasserstein [41, 7], et la classe des distances de transport composites (CTD) [178, 37]. Parmi ces dernières, seule la distance de Cauchy-Schwarz admet une formule close pour son évaluation ainsi que son gradient, tandis que les CTDs peuvent être facilement calculés en résolvant un problème de programmation linéaire [37].

**Conclusion** Chaque mesure de similarité présente ses avantages et inconvénients et il n'existe pas de consensus pour choisir la meilleure. Le choix dépend de la nature du problème, des contraintes techniques (temps de calcul, disponibilité des ressources) ainsi que les attentes de l'utilisateur. Nous verrons cela plus en détail dans les prochaines sections.

De plus, rappelons que le problème d'optimisation du modèle réduit (cf. équation 2.35) n'admet pas de solution exacte. Comme il est non-convexe, la convergence vers une solution optimale globale avec l'optimisation numérique n'est pas garantie. Pour ce faire, de nombreux algorithmes ont été proposés dans la littérature dans le but de fournir des solutions approximatives du problème de GMR avec une faible complexité de calcul, afin qu'ils puissent être utilisés dans des applications en temps réel. La plupart d'entre eux utilisent deux étapes successives : la réduction gloutonne et l'affinement [34].

### 2.3.3 Réduction gloutonne

Cette étape consiste à effectuer une réduction itérative du GMM original, jusqu'à ce que l'on obtienne un GMM avec le nombre prescrit de composantes, présentant un certain degré de similarité avec le GMM original et, espérons-le, non éloigné de la solution optimale de l'équation 2.35. Ainsi, le GMM réduit peut servir de bon point de départ pour des affinements ultérieurs. Les opérations de base impliquées à chaque itération sont l'élagage (pruning) et la fusion des composantes du GMM.

**Élagage** C'est le processus d'élimination d'une ou plusieurs composantes d'un GMM donné selon un certain critère. Les approches courantes consistent à éliminer les composantes selon des critères basés sur :

- les amplitudes des poids  $\pi$  (seuillage, k le plus petit, hors du centile prescrit,...),

— le coût en termes d'incrément de la mesure  $D$  choisie.

L'élagage selon des critères basés sur les poids est typique des contextes de traçage de cible [14]. Les composantes à supprimer peuvent être celles dont les poids sont inférieurs à un seuil fixe ou adaptatif, ou les composantes associées aux  $k$  plus petits poids,  $k$  étant fixe ou variant à chaque étape, ou encore celles associées aux plus petits poids dont la somme est inférieure à une valeur prescrite (typiquement 0,05). Cette approche est plus rapide que l'utilisation de critères basés sur les coûts, elle peut donc être utilisée pour accélérer la phase de réduction. D'un autre côté, elle peut conduire à une détérioration significative de l'approximation finale.

Les critères d'élagage basés sur les coûts visent à éliminer d'un GMM les composantes dont l'élimination entraînerait le plus faible accroissement de la mesure  $D$  choisie. Ainsi, pour un GMM de taille  $K_a$ , la mesure de dissimilarité doit être calculée  $K_a$  fois (un calcul pour chaque hypothèse d'élimination d'une composante), et ces calculs doivent être répétés à chaque étape d'élagage, jusqu'à ce que le nombre souhaité de composants  $K_b$  soit atteint. Il est clair que l'élagage selon les critères basés sur les coûts est généralement plus lent que l'élagage selon les critères basés sur les poids ; mais pour certaines mesures de dissimilarité, il peut améliorer le résultat de façon considérable. En revanche, il est important de noter que l'applicabilité des critères basés sur les coûts est limitée, car de nombreuses mesures de dissimilarité, comme le KLD, ne peuvent pas être calculées en forme fermée pour les GMMs. Dans de tels cas, des approximations [35] ou des bornes supérieures des mesures de dissimilarité [143] peuvent être utilisées, au prix d'une détérioration du résultat final. Enfin, quel que soit le critère utilisé, après chaque étape d'élagage, les poids doivent être normalisés, afin que leur somme soit toujours égale à un. En revanche, il a été avancé que la fusion des composants est préférable à l'élagage dans la plupart des cas [34, 171].

**Fusion** C'est le processus qui consiste à remplacer un sous-ensemble spécifique de composantes gaussiennes (un sous-mélange) d'un GMM donné par une seule composante gaussienne, selon un certain critère. La composante résultante est généralement le barycentre du sous-mélange [153], dont le poids est égal à la somme des poids des composantes du sous-mélange sélectionné. Formellement, en définissant  $p(x|\overline{\Theta}_{K_a})$  le sous-mélange non-normalisé d'ordre  $\overline{K}_a$  tel que  $\overline{K}_a < K_a$ , le problème du barycentre est définie comme suit :

$$(\hat{\mu}, \hat{\Sigma}) = \arg \min_{(\mu, \Sigma) \in \mathbb{R}^d \times \mathbb{S}^d} \sum_{i=1}^{\overline{K_a}} \bar{\pi}_i D(\mathcal{N}(\cdot | \bar{\mu}_i, \bar{\Sigma}_i) || \mathcal{N}(\cdot | \mu, \Sigma)). \quad (2.45)$$

tel que  $\mathbb{S}^d$  est l'ensemble des matrices carrées, positives et semi-définies sur  $\mathbb{R}^d$ .

Comme pour (2.35), le problème (2.45) est un problème d'optimisation non-convexe et n'admet pas de formule close pour la plupart des mesures D. En revanche, en utilisant la KLD (cf. équation 2.36) pour la mesure D, la solution du problème (2.45) est donnée par les équations suivantes :

$$\begin{aligned} \hat{\pi} &= \sum_{i=1}^{K_a} \bar{\pi}_i \\ \hat{\mu} &= \sum_{i=1}^{K_a} \bar{\pi}_i \bar{\mu}_i \\ \hat{\Sigma} &= \sum_{i=1}^{K_a} \bar{\pi}_i (\bar{\Sigma}_i + (\bar{\mu}_i - \hat{\mu})(\bar{\mu}_i - \hat{\mu})^T) \end{aligned} \quad (2.46)$$

Une formule close est aussi disponible pour la distance de Wasserstein [41]. Par contre, il n'existe pas de formules closes pour les autres mesures (ISE, NISE, etc...). Par conséquent, la plupart des algorithmes GMR effectuent des fusions en utilisant les barycentres de la KLD, par le biais des équations 2.46. Cependant, le calcul numérique du barycentre de l'ISE et NISE est possible de fait de la disponibilité d'une formule close du gradient ISE et NISE.

Le choix des composantes à fusionner dans un mélange peut être effectué à l'aide de critères de coût locaux ou globaux. Un critère local consiste à vérifier, à chaque étape, les dissimilarités entre toutes les paires de composants du GMM donné, puis à fusionner les paires présentant la plus faible dissimilarité (une variante de cette approche consiste à créer des clusters de plus de deux composants similaires et à effectuer la fusion du cluster le moins dissemblable). Un critère global consiste à vérifier les effets de toutes les fusions possibles de paires de composants sur la dissimilarité avec le GMM d'origine, puis à choisir de fusionner les composants qui donnent la plus faible dissimilarité.

**Combinaison de l'élagage et la fusion** À chaque étape d'un algorithme de réduction itératif de type glouton, soit l'élagage, soit la fusion est appliqué, le choix dépendant de

critères impliquant, en général, à la fois les poids et l'évaluation de la dissimilarité entre les GMM avant et après la réduction, comme dans [170, 5].

**Approche ascendante** Les étapes décrites précédemment appelées aussi "descendantes" supposent donc démarrer du GMM initial et visent à réduire ses composantes par (élagage ou fusion) jusqu'à l'obtention du nombre de composantes cible. Une autre approche "ascendante" proposée par [75] suggère initier un mélange réduit avec une gaussienne, puis ajouter de nouvelles composantes au mélange réduit en estimant la dissimilarité entre le GMM initial et réduit. En revanche, il a été prouvé que cette approche génère des résultats de moindre qualité par rapport à l'approche classique descendante [34, 65],

### 2.3.4 Affinement

Cette étape consiste à améliorer la solution générée par l'étape de réduction gloutonne afin de se rapprocher d'avantage de la solution optimale. La plupart des méthodes d'affinement dans la littérature peuvent être divisées dans les trois classes suivantes :

- optimisation [170, 171]
- clustering [146, 7]
- optimisation-clustering [178, 176]

Les procédures d'affinement consistent à utiliser des méthodes numériques standards pour résoudre le problème d'optimisation 2.35. L'ISE (équation 2.40) et le NISE (équation 2.44) conviennent parfaitement à cette approche, car il existe des formules closes pour leur calcul, ainsi que leurs gradients, ce qui n'est pas le cas pour la KLD (2.38). Bien évidemment, quelle que soit la méthode d'affinement qui est appliquée, la solution finale fournie par l'affinement converge vers est un minimum local proche du point de départ résultant de l'étape gloutonne. C'est pourquoi il est extrêmement important que la réduction gloutonne fournisse un bon point de départ pour l'affinement.

Lorsque la mesure  $D$  adoptée n'admet de formule close qu'entre paires de gaussiennes, comme le KLD et la distance de Wasserstein, le clustering peut être utilisé pour réduire la dissimilarité entre les GMM originaux et réduits. Dans certains cas [178, 176], le clustering et l'optimisation sont combinés ensemble.

### 2.3.5 GMRC

Proposé par [146], cet algorithme (Gaussian Mixture Reduction with Clustering) a été largement utilisé dans le cadre du GMR. Comme mentionné dans Crouse et al [34], ce dernier utilise les deux étapes successives : la réduction gloutonne et l'affinement (affinement en 2 sous-étapes successives).

**Réduction gloutonne** La réduction gloutonne est faite en utilisant l'algorithme de Runnalls [143] décrit dans l'algorithme 1 décrit ci-dessous. Ce dernier utilise une borne supérieure de la KLD (cf. équation 2.38) pour choisir les composantes à fusionner à chaque itération. L'algorithme 1 se termine lorsque le nombre de composantes cible a été atteint.

Les algorithmes de Salmond [144] et Pao [121] sont les mêmes que celui de Runnalls, mais ils utilisent une mesure de coût différente et il a été démontré qu'ils sont moins performants que la méthode de Runnalls [143]. L'algorithme de West [168] est similaire, sauf qu'au lieu de considérer toutes les fusions possibles à chaque étape, il ne considère que les fusions impliquant le composant ayant le plus petit poids. L'algorithme de West "amélioré", utilisé dans l'algorithme COWA [30], en est une variante plus performante. Ce dernier utilisé dans la réduction gloutonne donne l'algorithme GMRC-West. Les résultats de simulation issus des travaux de [146] montrent que le temps de traitement de l'algorithme GMRC-West est plus rapide que celui du GMRC original. Toutefois, les performances en terme de mesure de dissimilarité peuvent être moins bonnes que celles du GMRC original qui utilise l'algorithme de Runnalls dans cette étape [54].

**Premier affinement** L'objectif consiste à optimiser les paramètres du GMM réduit obtenu dans l'algorithme 1 afin de se rapprocher de la solution optimale du problème (2.35). Comme la KLD n'admet pas de formule close pour comparer des GMM, cette étape utilise une approche basée sur un clustering similaire à l'algorithme "k-moyennes" [100] pour affiner les paramètres du GMM réduit et est donnée par l'algorithme 2.

**Second affinement** Cette étape consiste à optimiser d'avantage les paramètres du GMM résultant lors du premier affinement. Pour ce faire, cette étape optimise la fonction objectif (équation 2.35) en adoptant l'ISE (équation 2.39) comme mesure de similarité. Comme l'ISE admet une formule close dans le cas d'une comparaison de deux GMM (équation 2.40), le second affinement consiste à optimiser les paramètres du GMM suivant en



---

**Algorithme 1 :** Réduction gloutonne : Algorithme de Runnals [143]

---

1. Définir le GMM réduit égal au GMM initial et le nombre de composantes cible  $K_b$
2. Le coût de la fusion des composants  $i$  et  $j$  dans le mélange actuel est la borne supérieure de l'augmentation de la divergence KL :

$$c_{ij} = \frac{1}{2}((\pi_i + \pi_j) \log(|\Sigma_{ij}|) - \pi_i \log(|\Sigma_i|) - \pi_j \log(|\Sigma_j|)) \quad (2.47)$$

où  $\Sigma_{ij}$  est la matrice de covariance du barycentre obtenu en fusionnant les composants  $i, j$  avec l'équation 2.46.

3. Fusionner les composants du mélange actuel ayant le  $c_{ij}$  le plus faible en utilisant l'équation 2.46. Puis, définir le mélange actuel (réduit) comme le mélange résultant de la fusion.
  4. Si le nombre de composantes cible  $K_b$  a été atteint, terminer. Sinon, retourner à l'étape 2.
- 

---

**Algorithme 2 :** Premier affinement (k-moyennes) [146]

---

1. Calculer une première estimation du GMM réduit  $g(x|\Theta_{K_b})$  en utilisant l'algorithme 1
  2. Pour chaque composante du mélange initial, déterminer quelle composante du mélange réduit est la plus proche.  
telle que la distance entre la  $i^{\text{ème}}$  composante du mélange original et la  $j^{\text{ème}}$  composante du mélange réduit est donnée par l'équation 2.37
  3. Les composantes du mélange original les plus proches de la même composante dans le mélange réduit forment un cluster.  
En utilisant l'équation 2.46, fusionner les composantes dans chaque cluster pour former le nouveau mélange réduit
  4. Si le nouveau mélange réduit est le même que l'ancien mélange réduit ou si le nombre maximal d'itérations s'est écoulé, terminer. Sinon, définir le mélange réduit comme le nouveau mélange réduit et retourner à l'étape 2.
- 

répétant successivement les deux étapes :

- Optimiser les vecteurs moyennes et matrices de covariances avec la descente du gradient.
- Optimiser les poids par programmation quadratique.

Le gradient de la fonction objectif (équation 2.40) est :

$$\nabla_{\Theta_{K_b}} D_{ISE}(f(\cdot|\Theta_{K_a})||g(\cdot|\Theta_{K_b})) = -2\nabla_{\Theta_{K_b}} J^{ab} + \nabla_{\Theta_{K_b}} J^{bb} \quad (2.48)$$

Pendant l'optimisation, les matrices de covariance doivent être symétriques et définies positives pour rester valides. Pour appliquer cette contrainte, Williams et al [171] ont effectué l'optimisation sur la racine carrée des covariances plutôt que sur les covariances elles-mêmes. C'est-à-dire que l'optimisation a été effectuée sur l'ensemble des  $L_i$  tels que :

$$\Sigma_i^b = L_i L_i^T \quad (2.49)$$

où le  $L_i$  initial peut être obtenu en prenant la matrice triangulaire inférieure de la décomposition de Cholesky [101] de l'estimation initiale de  $\Sigma_i$ . Les gradients sur  $\mu, L$  pour effectuer l'optimisation sont :

$$\begin{aligned} \nabla_{\mu_j^b} J^{bb} &= -2\pi_j^b \sum_{i=1}^{K_b} \pi_i^b (\Sigma_i^b + \Sigma_j^b)^{-1} (\mu_j^b - \mu_i^b) (H^{bb})_{ij} \\ \nabla_{\mu_j^b} J^{ab} &= -\pi_j^b \sum_{i=1}^{K_a} \pi_i^a (\Sigma_i^a + \Sigma_j^b)^{-1} (\mu_j^b - \mu_i^a) (H^{ab})_{ij} \\ \nabla_{L_j} J^{bb} &= 2 \sum_{i=1}^{K_b} \pi_i^b \pi_j^b (H^{bb})_{ij} (\Sigma_i^b + \Sigma_j^b)^{-1} ((\mu_i^b - \mu_j^b)(\mu_i^b - \mu_j^b)^T - (\Sigma_i^b + \Sigma_j^b)) (\Sigma_i^b + \Sigma_j^b)^{-1} L_j \\ \nabla_{L_j} J^{ab} &= \sum_{i=1}^{K_a} \pi_i^a \pi_j^b (H^{ab})_{ij} (\Sigma_i^a + \Sigma_j^b)^{-1} ((\mu_i^a - \mu_j^b)(\mu_i^a - \mu_j^b)^T - (\Sigma_i^a + \Sigma_j^b)) (\Sigma_i^a + \Sigma_j^b)^{-1} L_j \end{aligned} \quad (2.50)$$

L'optimisation ne doit pas être effectuée sur chaque élément des matrices  $L_i$ , mais seulement sur les éléments triangulaires inférieurs de chaque  $L_i$ , les éléments triangulaires supérieurs étant fixés à zéro.

Pour l'optimisation des poids, il faut s'assurer que tous les poids sont positifs et que leur somme soit égale à 1. En revanche, il a été prouvé par simulation que l'optimisation des poids suivant la descente du gradient ne garantissait pas les contraintes. En pratique, il faut ajouter une contrainte supplémentaire pour s'en assurer. Pour ce faire, l'expression 2.40 est optimisée suivant les poids en gardant les vecteurs moyennes et matrices de covariances fixés. L'optimisation devient alors un problème de programmation quadratique convexe, et (en omettant le terme constant de 2.40 et en multipliant tout par 1/2) peut

être formulé comme suit :

$$\begin{aligned} \min_{\pi^b} \quad & \frac{1}{2} \pi^b H^{bb} \pi^b - \pi^a H^{ab} \pi^b \\ \text{sujet à} \quad & \sum_{i=1}^{K_b} \pi_i^b = 1 \\ & -\pi_i^b \leq 0; \quad \forall i \in \{1, \dots, K_b\} \end{aligned} \quad (2.51)$$

Comme  $H^{bb}$  est définie positive [34], il s'agit d'un problème de programmation quadratique convexe qui peut donc être réduit à un problème de programmation linéaire et résolu exactement en un temps polynomial [11, 49]. La contrainte d'inégalité empêche le problème d'avoir une solution explicite.

Dans l'ensemble, étant donné l'estimation des paramètres de la deuxième étape de la GMRC, une optimisation de la descente du gradient peut être effectuée sur les moyennes et les paramètres non nuls de la décomposition de Cholesky des matrices de covariance, ce qui permet, pour chaque itération, de trouver les poids de mélange globalement optimaux par programmation quadratique.

## 2.4 Conclusion

Dans cette partie, les notions de base associées aux GMM ont été fournies. Ensuite, nous les mécanismes de décisions avec rejet ont été présentés, permettant de détecter si des observations sont éloignées des gaussiennes apprises (rejet de distance), ou celles qui sont à la frontière de plusieurs gaussiennes simultanément (rejet d'ambiguïté). Enfin, une revue de littérature sur la réduction des GMM a été fournie. Cette dernière permet de simplifier un GMM en ne retenant que les gaussiennes pertinentes et en fusionnant les gaussiennes similaires. Ce mécanisme permet de réguler le nombre de gaussiennes, notamment pour les processus qui génèrent itérativement de nouvelles composantes.

Le GMM est un approximateur universel de densité, ce qui le rend idéal pour la classification des données en production industrielle, notamment lorsqu'elles sont déséquilibrées. De plus, les deux techniques (décision de rejet et réduction de modèles) sont très utiles pour détecter de nouvelles classes et de garantir une stabilité du GMM au cours du temps. Par conséquent, une architecture basée sur un GMM entourée des 2 techniques est pro-

metteuse pour effectuer de la surveillance en temps réel. Enfin, Il convient de rappeler que ces techniques n'ont pas été appliquées pour la détection de défaut dans la surveillance de processus de fabrication.

En revanche, plusieurs contextes opératoires différents peuvent exister en production. De plus, plusieurs outils ou machines avec des cycles de vie différents, peuvent être utilisés en même temps. Ces différents paramètres peuvent affecter les signatures de défauts dans le sens où ces dernières peuvent être différentes suivant ces différents paramètres opératoires. En conséquence, la surveillance de tous les processus de production avec un seul GMM devient irréaliste. En outre, les approches de surveillance basées sur les données montrent souvent leurs limites sur les productions industrielles, notamment en présence de données complexes. D'un autre côté, la formalisation et l'intégration des connaissances en interagissant avec un expert du domaine est souhaitable afin d'apporter une confiance supplémentaire dans l'outil de surveillance. De ce fait, un outil permettant d'organiser toutes les informations apprises suivant les différents paramètres de façon efficace est fortement requis. Un outil élégant permettant de le faire est une base de connaissance modélisée par une ontologie.



# ONTOLOGIE

---

Dans le chapitre précédent, nous avons montré que la complexité des sites de production rend l'utilisation d'un seul modèle d'apprentissage inefficace quelle que soit sa généralité. En effet, la multitude de paramètres de surveillance -donne lieu à- des contextes opératoires divers, qui ne peuvent être capturés par un seul modèle. De ce fait, une ontologie peut gérer ce problème du fait de sa capacité à gérer les connaissances.

Dans de nombreux cas, les ontologies disponibles fournissent des connaissances de haut niveau pour le même domaine étudié [60]. Le recours aux ontologies est essentiellement dû à leur capacité à capturer la sémantique de l'expertise du domaine. Ainsi, de nombreuses solutions ontologiques ont été mises en œuvre dans plusieurs applications réelles dans des domaines aussi variés que la traduction en langage naturel [85], la médecine [61], le commerce électronique [92] et la bio-informatique [6].

Par conséquent, les différentes informations apprises par les modèles d'apprentissage peuvent être enregistrées en tant que connaissance dans l'ontologie. Réciproquement, l'exploitation de l'ontologie peut inférer de nouvelles connaissances qui peuvent améliorer les modèles d'apprentissage.

Ce chapitre est organisé comme suit : une introduction formelle l'ontologie est présentée dans la section 3.1. La section 3.2 présente les cas d'utilisation d'une ontologie dans le domaine de diagnostic de défauts. Enfin, des approches hybrides combinant l'apprentissage automatique avec des ontologie sont présentées dans la section 3.3 en discutant leurs limites.

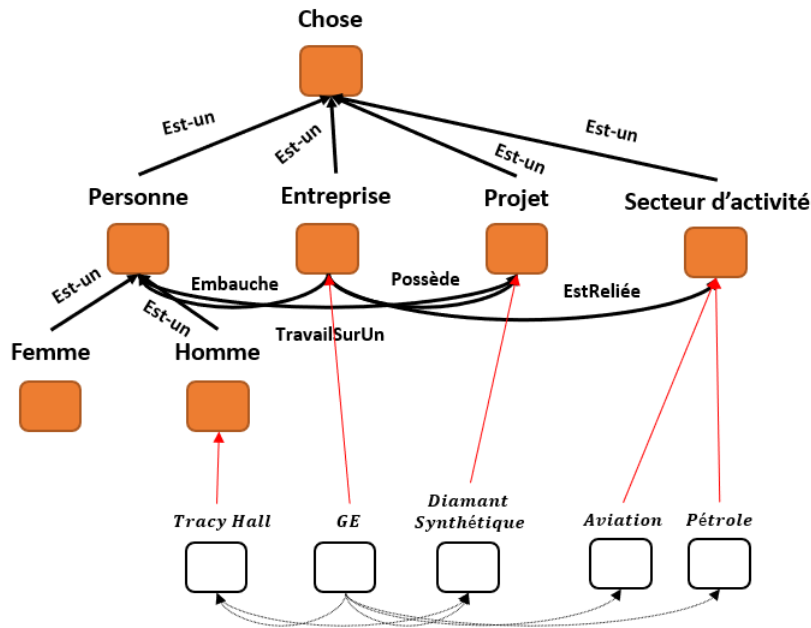


FIGURE 3.1 – Un exemple d'une ontologie : concepts, relations sémantiques et instances

## 3.1 Notions de base

Un système à base de connaissances (KB) fournit un cadre de raisonnement cohérent doté d'un moteur d'inférence qui raisonne de manière déductive sur un langage logique. L'ontologie est l'un de ces types de systèmes basés sur la connaissance sémantique. Il existe différentes définitions dans la littérature de ce que devrait être une ontologie. La plus acceptée a été donnée par Gruber et al [67], stipulant qu'une ontologie est une spécification explicite d'une conceptualisation. La "conceptualisation", ici, fait référence à un modèle abstrait d'un certain phénomène ayant une réalité en identifiant ses concepts pertinents. Le mot "explicite" signifie que tous les concepts utilisés et les contraintes sur leur utilisation sont explicitement définis.

### 3.1.1 Définition

En théorie, une ontologie peut être définie comme suit :

- Un ensemble de concepts ou de classes  $\mathcal{C} = \{C_1, \dots, C_n\}$  structuré au moyen d'une hiérarchie taxonomique (is-a) et partonomique (part-of)  $\mathcal{H}$ ,
- Des propriétés ou attributs des concepts,

- Des relations sémantiques entre les concepts ( $\mathcal{R}_c : C_i \times C_j$ ), où  $C_i, C_j$  sont respectivement le domaine et l'étendue de la relation sémantique, l'ensemble des concepts et des relations sémantiques forment la boîte terminologique (nommé aussi "T-Box"),
- Un ensemble d'instances de concepts (resp. de relations)  $\mathcal{I}$  formant la boîte d'assertion (nommé aussi "A-box"),
- Un ensemble d'axiomes formels  $\mathcal{A} = \langle c_{ik}, c_{jm}, v_n \rangle$  avec  $c_{ik}, c_{jm} \in \mathcal{I}$  et  $v_n \in \mathcal{V}$  (c'est-à-dire un ensemble de contraintes comme doit, ne doit pas, devrait, ne devrait pas, etc.)

Un exemple d'une ontologie est illustré sur la figure 3.1. Les carrés oranges représentent les concepts et les flèches annotées sont les relations sémantiques entre concepts. Les carrés blancs sont des exemples d'instances des concepts.

### 3.1.2 Les catégories de l'ontologie

Les ontologies peuvent exister à plusieurs niveaux d'abstraction. Plus précisément, trois catégories d'ontologies sont distinguées : les ontologies de niveau supérieur, de niveau intermédiaire et de domaine (voir figure 3.2).

- Une ontologie supérieure (ou de base) est une ontologie de premier niveau, indépendante du domaine, à partir de laquelle des ontologies plus spécifiques au domaine peuvent être dérivées. Les concepts exprimés dans une telle ontologie sont censés être méta, génériques et abstraits afin de garantir l'expressivité pour un large éventail de domaines.
- Une ontologie de niveau intermédiaire est conçue spécifiquement pour servir d'interface entre les concepts de haut niveau définis dans l'ontologie supérieure et les concepts de bas niveau spécifiés dans une ontologie de domaine. En d'autres termes, une ontologie de niveau intermédiaire est une ontologie supérieure pour un domaine spécifique.
- Une ontologie de domaine (ou spécifique à un domaine) modélise un domaine spécifique de connaissances, ou une partie du monde. Elle représente les significations particulières des termes tels qu'ils s'appliquent à ce domaine. Les ontologies de domaine peuvent également étendre les concepts définis dans les ontologies de niveau intermédiaire et supérieur.



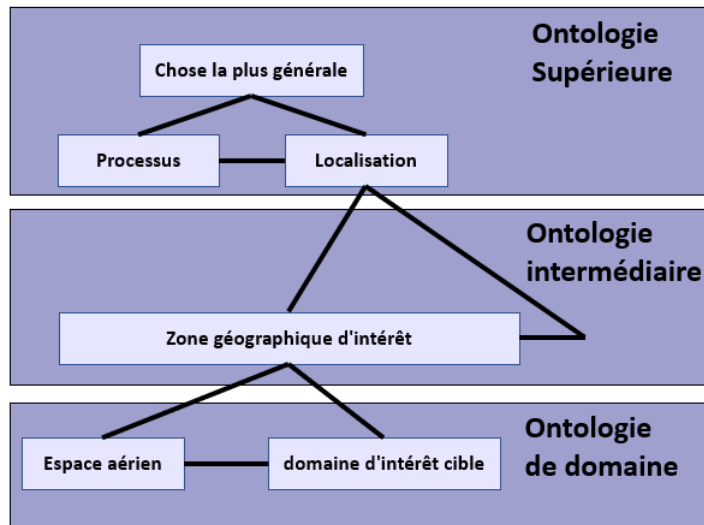


FIGURE 3.2 – Les catégories de l'ontologie

### 3.1.3 Les utilisations d'ontologie

Dans cette section, les motivations de base pour utiliser et développer des ontologies sont fournies. Pour les tâches qui font appel à un domaine spécifique de connaissances, il peut être plus adéquat d'utiliser des ontologies de domaine plutôt que des ontologies plus générales. Ce faisant, l'espace des utilisations des ontologies est divisé dans trois catégories.

L'interopérabilité est souvent le premier espace d'utilisation des ontologies. En effet, de nombreuses applications basées sur les ontologies abordent les questions d'interopérabilité dans lesquelles différents systèmes de gestion de l'information sont déployés et différents utilisateurs de systèmes doivent échanger des données en utilisant divers outils logiciels [133]. La principale contribution au recours aux ontologies dans des domaines tels que la modélisation d'entreprise et les architectures multi-agents est la création d'une compréhension partagée de domaines communs permettant aux applications de s'accorder sur les termes qu'elles utilisent lorsqu'elles communiquent. Ainsi, les ontologies, si elles sont partagées entre les applications inter-opérantes, permettent un échange de données sémantiques entre ces outils.

L'ingénierie des systèmes figure dans l'espace d'utilisation de l'ontologie. En effet, une compréhension partagée et cohérente des problèmes et des tâches à accomplir peut aider à la spécification d'un projet de génie logiciel. Ainsi, des ontologies de génie logiciel sont développées afin de représenter et de communiquer les connaissances en génie logiciel. Le

développement d'une telle "ontologie de domaine du génie logiciel" nous permettra de :

- Partager et réutiliser toutes les connaissances accumulées jusqu'à présent dans le domaine du génie logiciel,
- Ouvrir de nouvelles voies à l'interprétation automatique de ces connaissances.

Par exemple, le projet SWEBOK (Software Engineering Body of Knowledge [17]), est le résultat d'un grand effort d'extraction de connaissances par le biais d'acquisition et de structuration de documents très divers (articles scientifiques, actes de congrès, livres, chapitres, rapports techniques, normes techniques), et de connaissances de base provenant d'experts du domaine, de consultants et de chercheurs. L'équipe du projet SWEBOK a établi le projet avec cinq objectifs : 1) caractériser le contenu de la discipline du génie logiciel ; 2) fournir un accès thématique au corpus de connaissances du génie logiciel ; 3) promouvoir une vision cohérente du génie logiciel dans le monde entier. 4) clarifier la place et fixer les limites du génie logiciel par rapport à d'autres disciplines telles que l'informatique, la gestion de projet, le génie informatique et les mathématiques ; 5) fournir une base pour le développement de programmes d'études et de matériel de certification individuelle.

Enfin, l'ontologie peut être utilisée à des fins de communication. Rappelons que les ontologies réduisent considérablement la confusion terminologique en fournissant un cadre unifié au sein d'une organisation. De cette façon, les ontologies permettent une compréhension partagée et une communication entre des personnes ayant des besoins et des points de vue différents découlant de leurs contextes particuliers.

### 3.1.4 Évolution de l'ontologie

Un point critique dans l'application des ontologies aux problèmes du monde réel est que les domaines évoluent rapidement (nouveaux concepts, concepts changeant de signification, nouvelles relations, nouveaux axiomes, etc.) et que les besoins des utilisateurs évoluent également. Par conséquent, les ontologies correspondantes doivent elles aussi évoluer.

L'évolution de l'ontologie est l'adaptation opportune de l'ontologie en réponse à un certain besoin. Plusieurs raisons de changer d'ontologie ont été identifiées dans la littérature :

- Un changement dynamique dans le domaine modélisé [156].

- Un changement de la perspective sous laquelle le domaine est vu [115].
- Une découverte d'un défaut de conception ou d'un changement d'orientation de la conception initiale [126].
- Une nécessité d'intégrer des fonctionnalités supplémentaires en fonction de l'évolution des besoins de l'utilisateur [69].
- Une disponibilité de nouvelles informations, auparavant inconnues, comme l'apparition de différentes caractéristiques du domaine [71].

Le processus d'évolution fait passer l'ontologie d'un état cohérent à un autre [23] et peut être de deux types :

1. Peuplement de l'ontologie : Lorsque de nouvelles instances de concept(s) déjà présents dans l'ontologie sont obtenues. Seules les nouvelles instances sont ajoutées et l'ontologie est peuplée.
2. Enrichissement d'ontologie : Qui consiste à mettre à jour (ajouter ou modifier) les concepts, propriétés et relations d'une ontologie donnée.

Les changements les plus courants [23] peuvent être résumés comme suit :

- Ajout de nouveaux concepts : Il s'agit du changement le plus courant dans toute ontologie. De nouveaux concepts apparaissent et doivent être intégrés dans la hiérarchie des concepts déjà existante.
- Modifier la hiérarchie des concepts : Dans ce cas, le concept en question peut avoir une position hiérarchique différente de celle du concept existant.
- Modification des propriétés du concept : Lorsque le concept en question est déjà présent dans l'ontologie mais que ses propriétés sont différentes de celles de l'existant.
- Modification des restrictions du concept : Dans ce cas, le concept en question a des restrictions qui sont différentes de celles associées aux concepts existants.
- Ajout de nouvelles relations (taxonomiques ou non) entre des concepts existants.

## 3.2 Ontologie pour le diagnostic de défauts

Dans le diagnostic de défauts (détection, isolation et correction) basé sur l'ontologie, la manière d'utiliser l'ontologie de diagnostic de défaut est l'un des facteurs décisifs pour l'efficacité et l'effet de cette méthode. À l'heure actuelle, l'utilisation de l'ontologie de

diagnostic des défauts peut être divisée en trois catégories : le partage du vocabulaire du domaine, le raisonnement sur les causes des défauts et la récupération des informations ou le calcul des similarités [99].

### 3.2.1 Partage du vocabulaire

Il est nécessaire de partager les vocabulaires de diagnostic de pannes avant de partager les connaissances du domaine et d'échanger des données lorsque le diagnostic de pannes est effectué dans un environnement distribué. L'ontologie du diagnostic de défaut facilite le partage du vocabulaire du domaine dans les environnements distribués.

Dans un système de diagnostic de pannes composé de plusieurs agents, chaque agent doit comprendre et partager le vocabulaire du domaine du diagnostic de pannes afin d'échanger des informations sur la panne. L'ontologie est utilisée comme un outil et un moyen de gérer les connaissances inter ou intra-domaines, afin d'assurer la communalité et la cohérence des modèles d'information. En partageant les vocabulaires de l'ontologie de diagnostic des défauts, les agents multiples peuvent coordonner leurs actions lorsqu'ils utilisent des réseaux bayésiens pour le diagnostic des défauts [24], ou peuvent coopérer et négocier avec les utilisateurs et d'autres agents pour accomplir des tâches de diagnostic des défauts [3].

Dans les données historiques de diagnostic de pannes collectées dans des environnements distribués, il y a souvent des cas où les mêmes composantes sont exprimées par des vocabulaires différents et incohérents. L'ontologie de diagnostic de pannes peut alors être utilisée pour nettoyer les données bruyantes [131] et collecter des données fines [132]. Elle peut également être utilisée pour améliorer la qualité des connaissances extraites des rapports de réparation collectés à partir de sources distribuées ou différentes [130].

### 3.2.2 Raisonnement sur les causes de défauts

En tant que base de connaissances du domaine, l'ontologie de diagnostic des défauts peut être utilisée directement ou indirectement pour le raisonnement afin de découvrir les causes des défauts et de fournir des propositions de réparation ou de maintenance.

L'ontologie de diagnostic de pannes peut être combinée avec des règles pour être exploitée par des moteurs d'inférence, visant à trouver les causes des pannes et à déduire

des propositions de réparation ou de maintenance. Ainsi, des travaux qui visent à identifier les causes de défauts avec différentes techniques peuvent être exploitées par une ontologie en aval [106]. Par exemple, en utilisant l'ontologie et les règles pour représenter la connaissance sur le diagnostic de défauts des machines tournantes, le raisonneur de l'ontologie peut dériver la cause du défaut de la machine tournante [31]. Après avoir modélisé les connaissances extraites de l'AMDEC à l'aide de l'ontologie, le raisonneur de l'ontologie peut aider le personnel de maintenance à trouver rapidement les causes des défauts des éoliennes et à choisir une solution appropriée pour réparer les éoliennes [178]. Dans l'exécution du diagnostic des défauts pour les tracteurs, l'ontologie est utilisée en conjonction avec des règles pour déduire la cause du défaut, lorsqu'il y a un manque de cas de défauts similaires et que le raisonnement basé sur les cas est insuffisant pour le diagnostic des défauts [173]. L'ontologie de diagnostic des pannes est le plus souvent représentée par un langage Web ontologique et combinée avec des règles représentées par un langage de règles du Web sémantique (SWRL), lorsqu'elle est utilisée pour un raisonnement direct [145, 178, 173]. Cela permet de déduire la cause du défaut en utilisant des moteurs d'inférence d'ontologie couramment utilisés tels que JESS, Racer, Jena et Pellet.

### 3.2.3 Calcul des similarités

L'ontologie de diagnostic de défaut décrit les informations sur l'objet de diagnostic et lui crée un profil spécifique. Par conséquent, la cause du défaut et les mesures de maintenance peuvent être trouvées par le calcul de similarité ou le raisonnement à partir de cas (CBR) lors du diagnostic du défaut, à condition qu'il existe un cas historique similaire.

L'ontologie de diagnostic de défaut peut être utilisée pour décrire le cas de diagnostic de défaut, calculer la similarité et stocker le cas, fournissant ainsi un moyen de récupérer le cas pour le diagnostic de défaut [173, 180, 43].

Puisque l'ontologie de diagnostic de défaut peut représenter des informations hétérogènes de sources multiples, elle est utilisée pour fournir plus d'informations connexes pour retrouver des cas similaires de diagnostic de défaut [107]. Dans un algorithme de clustering pour extraire les triplés des mots-clés fréquents <symptômes, composantes, meilleures pratiques de maintenance>, l'ontologie de diagnostic de défaut est utilisée pour représenter les mots-clés dans le rapport du technicien de réparation. L'ontologie est également

utilisée dans le calcul des métriques de similarité des cas de diagnostic de défauts, ce qui donne de meilleurs résultats de clustering [130].

### 3.2.4 Synthèse

Grâce à l'exploitation des connaissances existantes sur le diagnostic des défauts, l'ontologie est capable de découvrir des connaissances inconnues pour traiter de nouvelles situations complexes. Deux facteurs principaux affectent les performances du diagnostic de défauts basé sur l'ontologie : la quantité de connaissances du domaine et la capacité à appliquer ces connaissances pour résoudre des problèmes pratiques. Par conséquent, il est nécessaire d'obtenir plus de connaissances et d'utiliser ces connaissances de manière appropriée [99].

Toutefois, l'inefficacité de l'acquisition de connaissances massives limite la constitution d'une ontologie efficace, ce qui entrave la poursuite du développement. De plus, les recherches actuelles se concentrent sur la modélisation des connaissances a priori et négligent l'utilisation des données d'exploitation. Il en résulte un développement partiel d'une ontologie efficace permettant de fournir un système de gestion défauts [181].

Récemment, des approches "hybrides" combinant des techniques de l'apprentissage automatique et l'ontologie ont été proposées afin de lever ce problème.

## 3.3 Approches hybrides

À l'ère du Big Data, la découverte de connaissances à partir de données est une tendance qui prend de l'ampleur. En effet, l'acquisition des connaissances par la fouille de données [56] pour la modélisation de l'ontologie de diagnostic de défaut est également nécessaire. L'application des techniques d'apprentissage automatique dans l'ingénierie des connaissances peut étendre la base de connaissances et améliorer la qualité des connaissances, comme l'apprentissage des concepts et les relations sémantiques entre les concepts.

À l'heure actuelle, des tentatives ont été faites pour combiner l'apprentissage automatique et l'ontologie. Nous classons ces dernières en 2 catégories : approches d'apprentissage automatique pour l'ontologie (section 3.3.1) et approches de l'ontologie pour l'apprentissage automatique (section 3.3.2).

### 3.3.1 L'apprentissage automatique pour l'ontologie

Dans la littérature, très peu de travaux qui utilisent l'apprentissage automatique pour améliorer la qualité de l'ontologie de diagnostic de défaut existent. En revanche, quelques travaux utilisant cette approche pour une autre tâche existent. Par exemple, dans la détection des cyberattaques, l'apprentissage profond a été utilisé dans un premier sur des fichiers log afin d'apprendre des représentations latentes [179]. Ensuite, ces représentations serviront à définir des concepts et relations sémantiques d'une ontologie. Enfin, l'ontologie est utilisée pour définir les caractéristiques des nouveaux fichiers log qui vont être alimenté à un classifieur (XGBoost).

Une autre utilisation du ML est pour apprendre des règles logiques qui vont ensuite être utilisées par le raisonneur de l'ontologie. À cet effet, une approche par la fouille de données est souvent appliquée sur des données d'apprentissage pour générer ces règles [39, 40]. Par exemple, les arbres de décisions ont été utilisés pour apprendre des règles logiques sous le langage du *Semantic Web Rule Language* (SWRL) pour la gestion de la consommation électriques dans les cités intelligentes [102, 8]. En outre, nous retrouvons deux travaux utilisant cette approche pour le diagnostic de défauts. En effet, les *c*-moyennes floues ont été utilisé pour apprendre des règles qui ont été formulées ensuite sous le langage SWRL dans l'ontologie de diagnostic de défauts [21, 159]. Enfin, des règles métier extraites des connaissances des expert ainsi que de la fouille de données ont été intégrées dans un jumeau numérique afin de caractériser le comportement réel du système physique tout au long de sa phase de fonctionnement [89]. Précisément, ces règles permettent de détecter les défauts et de décrypter le contexte opérationnel dans l'usage.

Toujours dans le cadre de la diagnostic de défauts, une étude a été réalisée pour identifier les causes des défauts détaillées ainsi que le programme de maintenance adéquat [181]. Pour ce faire, les défauts sont détectés à travers un apprentissage non supervisé (chaîne de Markov cachées). Ensuite, les défauts sont instanciés dans l'ontologie à travers un lien sémantique. Enfin, les causes racines ainsi que le programme de maintenance sont inférés à travers de règle SWRL préalablement définies.

### 3.3.2 L'ontologie pour l'apprentissage automatique

Dans le cadre de l'application des ontologies pour l'amélioration des algorithmes d'apprentissage automatique, nous retrouvons peu de travaux dans le cadre de la détection

de défauts. En effet, un premier travail concerne la surveillance du processus de soudage dans un cas d'études sur Bosch [158]. Pour ce faire, l'ontologie a été utilisée pour améliorer la communication et l'interopérabilité des différents acteurs de la surveillance (managers, data-scientistes, développeurs). En effet, dans un premier temps, des ontologies ont été proposées pour chacun des acteurs afin de simplifier la communication. Ensuite, d'autres ontologies ont été proposées afin d'associer un langage commun entre les différentes sources de données. Ce mécanisme a permis d'améliorer considérablement la gestion de différentes sources de données hétérogènes. Enfin, des caractéristiques prédéfinies comme concepts dans l'ontologie permet d'extraire les informations pertinentes des différentes sources de données. Ce travail a permis d'améliorer considérablement tout le pipeline de la surveillance process. Par ailleurs, une ontologie agrégeant différentes sources de données hétérogènes afin d'être exploitées par des algorithmes de fouille de données a été proposée dans un contexte d'usinage [120].

Un autre travail concerne la prédiction des défauts dans les éoliennes [172]. Pour ce faire, l'ontologie a été utilisée pour pré-traiter les données et d'extraire des caractéristiques définies comme concepts dans l'ontologie. Ensuite, ces caractéristiques sont utilisées pour apprendre un réseau de neurones profond (CNN et LSTM) afin de détecter les défauts. Cette approche trouve son point fort dans l'utilisation des connaissances métiers formulées par une ontologie pour simplifier le pré-traitement des données.

### 3.3.3 Quelques critiques des approches citées

Les approches précédentes ont permis de surmonter des difficultés souvent rencontrées dans la réalité. En effet, cela a permis de s'approcher d'avantage vers une solution implémentable en milieu industriel. En revanche, plusieurs aspects qui peuvent limiter l'application réelle n'ont pas été abordés. En effet, toutes les approches mentionnées donnent des résultats prometteurs mais ponctuels. En réalité, il faudrait que de nouveaux concepts, relations (i.e connaissances) puissent être appris au cours du temps. En parallèle, plusieurs connaissances peuvent devenir obsolètes et peuvent affecter négativement le système de surveillance. Cet aspect d'ajout et/ou de remise en cause de connaissances dans le temps (i.e maintenabilité de la base de connaissance) est un problème important qui n'a pas été abordé dans la plupart des travaux.

Concernant le domaine de la détection de défauts, à notre connaissance, aucun travail



n'a été proposé pour la détection itérative de défauts. La plupart se concentrent plus dans la dernière partie du diagnostic des défauts (cause et mesure de correction) et délaissent la partie de détection, avec des approches basiques où les défauts sont connus et appris de manière supervisée et ponctuelle. En revanche, comme les données arrivent en flux continue en industrie, ces dernières peuvent comprendre des observations de nouveaux défauts, ou une dérive des défauts déjà appris. Ainsi, un modèle de classification classique, qui a été entraîné sur un jeu de données initial, pour un nombre donné de défauts connus, ne pourra pas détecter ces nouveaux comportements et les classifera à tort dans des classes de défauts connus. De ce fait, il convient de proposer un mécanisme de surveillance de processus de fabrication permettant de détecter de nouveaux comportements (défauts ou dérive) dans le temps.

Par ailleurs, peu de travaux ont été abordés pour la gestion de multiples contextes opérationnels industriels. En effet, comme évoqué dans la section 2.4, plusieurs signatures différentes pour le même défaut peuvent être apprises suivant des conditions opératoires différentes et les outils/machines différentes, etc, et bien d'autres paramètres (contextes). Ainsi, il existe un fort besoin industriel pour découvrir, capturer, et gérer de nouveaux défauts suivant différents contextes de production.

Enfin, nous notons que la coopération entre l'apprentissage automatique et la gestion de connaissance, via une ontologie, dans toutes les contributions précédentes n'est bénéfique que dans un sens (c'est-à-dire que l'apprentissage automatique aide à améliorer l'utilisation de l'ontologie ou l'inverse, mais pas les deux). Une direction possible de la recherche est de développer des stratégies itératives qui proposent une réelle coopération et enrichissement dans les deux sens.

### **3.4 Conclusion**

Dans ce chapitre, les notions de base relatives aux ontologies ont été présentées. Puis, l'usage de ces ontologies dans le domaine de diagnostic de défauts a été illustré. Quelques travaux combinant l'apprentissage automatique avec des ontologies ont été fournis, tout en adressant leurs limites. Arrivés à la fin de la première partie, il convient de rappeler les principaux manquements que nous avons identifiées dans l'état de l'art.

Premièrement, les données issues des opérations de fabrication sont non-labélisées et déséquilibrées, ce qui nécessite des modèles de surveillance process non-supervisés capables

de gérer des données déséquilibrées. Ensuite, les modèles de surveillance process proposés dans la littérature n'incluent pas l'option de détecter et d'intégrer de nouveaux défauts au cours du process. De plus, il n'existe pas de consensus quant au meilleur modèle à utiliser dans la fabrication process, et de ce fait, il existe un fort besoin d'utiliser des modèles génériques. En outre, la surveillance et la réduction du sur-apprentissage des modèles de surveillance process dans le temps n'a pas été explicitement proposée. Enfin, l'intégration des connaissances métiers (de l'expert du domaine) dans le pipeline de ML est rarement évoquée. Cette dernière apporte une confiance très importante aux industriels afin d'envisager une implémentation industrielle.

Deuxièmement, nous avons affirmé qu'un changement des paramètres opératoires pourrait affecter la performance du modèle de surveillance. Cela est expliqué par le fait que l'hypothèse de *l'indépendance et l'identité distribuées (i.e. iid)* des données sur laquelle repose la plupart des modèles d'ML n'est potentiellement plus respectée. De ce fait, afin de ne pas utiliser des ANN profonds (Adaptation au Domaine) considérés comme des boîtes-noires, nous avons affirmé qu'une base de connaissance modélisée par une ontologie pourrait gérer, capturer et exploiter les connaissances issues des différents paramètres opératoires, et ainsi, permettant la gestion de différents contextes industriels.

Enfin, nous avons montré que l'utilisation des ontologies pour améliorer les performances des techniques de ML a été rarement discuté dans la littérature. En effet, la tendance a été plus centrée sur l'utilisation des techniques de ML pour alimenter une ontologie pour des fins de diagnostic de défauts. En revanche, l'exploitation des connaissances de l'ontologie pourrait améliorer les performances des techniques de ML, et ainsi, fournir un système de surveillance process plus performant.

Les prochains chapitres seront dédiés à la présentation de nos contributions adressant les limites énoncées.



DEUXIÈME PARTIE

# Contributions

---

# DÉCOUVERTE ITÉRATIVE DE CONNAISSANCES POUR LA DÉTECTION DE DÉFAUTS

---

Dans ce chapitre, une nouvelle architecture hybride Ontologie-ML est proposée pour la découverte itérative de connaissances dans la détection de défauts. Ce chapitre est organisé comme suit : notre architecture pour la découverte itérative de connaissances dédiée à la détection de défauts est présentée dans la section 4.1. La section 4.2 présente les expériences sur les jeux de données de l'UCI. Enfin, les conclusions et les perspectives sont tirées dans la section 4.3.

## 4.1 L'architecture proposée

Dans une entreprise de fabrication, une grande variété de machines, de processus, de stratégies de fabrication ou de conditions opératoires peuvent exister. Cette diversité rend difficile l'établissement d'un modèle mathématique unique qui puisse effectuer une surveillance efficace pour toutes ces conditions. En effet, la forme caractéristique (signatures) du signal du capteur généré lors de l'apparition d'un défaut au cours d'une opération de fabrication peut permettre sa détection. Cette signature sera spécifique à un type de défaut et de capteur. En outre, cette dernière peut différer d'un contexte (c'est-à-dire d'une machine, d'un outil, d'une stratégie, etc) donné à un autre, même si elles décrivent le même défaut. De plus, de nouveaux défauts inconnus, ou qui n'était pas présent dans le jeu de données d'apprentissage du modèle de surveillance, peuvent apparaître ultérieurement au cours de la production. Ces nouveaux défauts ne peuvent pas être détectés par les approches ML classiques (qui sont souvent entraînées de manière supervisée) et

nécessitent souvent de ré-entraîner le modèle en incluant les nouveaux défauts. En outre, en raison de la variété des contextes et de la complexité des modèles, cette approche peut rapidement conduire à de nombreux modèles spécifiques à chaque configuration, plus ou moins bien entraînés, et sans possibilité de transferts des résultats. Enfin, cette approche peut surcharger les ressources computationnelles, ce qui se traduit par une faible efficacité de la surveillance. Pour surmonter ces contraintes, un mécanisme de découverte de défaut potentiellement inconnu par ML est proposé dans la section 4.1.1. Il est associé à un modèle de connaissance basé sur une ontologie qui permet de gérer les signatures des défauts. L'ontologie s'enrichira progressivement et permettra, par des mécanismes de généralisation, de transférer ces connaissances à d'autres contextes.

#### 4.1.1 Architecture globale de détection de défaut inconnu

Notre objectif est de proposer une méthode basée sur une ontologie qui soit dédiée à la surveillance en temps réel et à la découverte de nouveaux défauts dans différents contextes de fabrication. Pour ce faire, une ontologie de détection des défauts (ODD) est présentée. Elle s'appuie sur quatre concepts fondamentaux :  $\mathcal{C} = \{\text{Contexte}, \text{État}, \text{Signature}, \text{Occurrence}\}$ . La hiérarchie  $\mathcal{H}$  est présentée dans la figure 4.1. Les carrés oranges représentent les classes et ceux en blancs un exemple de leurs instances. Les instances numérotées par un exposant dans la figure 4.1 indiquent qu'un concept particulier (par exemple le concept Occurrence) peut avoir plus d'une instance éventuellement ( $Occ^1, Occ^2, \dots$ ).

Les définitions détaillées des classes sont présentées dans le Tableau 4.1. En effet, le concept Contexte regroupe tous les éléments d'une usine de fabrication ayant un impact

TABLE 4.1 – Ontologie pour la Détection de Défauts : définitions des concepts

Concept	Définition
Contexte	Conditions exhaustives dans lesquelles est effectuée une opération de fabrication
État	Estimation de la qualité résultant d'une opération de fabrication (correcte ou présence d'un certain type de défaut)
Signature	Les éléments distinctifs d'un signal de capteur qui caractérisent un État dans un certain Contexte
Occurrence	Classe associative qui se réfère au nombre d'opérations de fabrication classées par association de signature dans un État donné pour un certain Contexte.

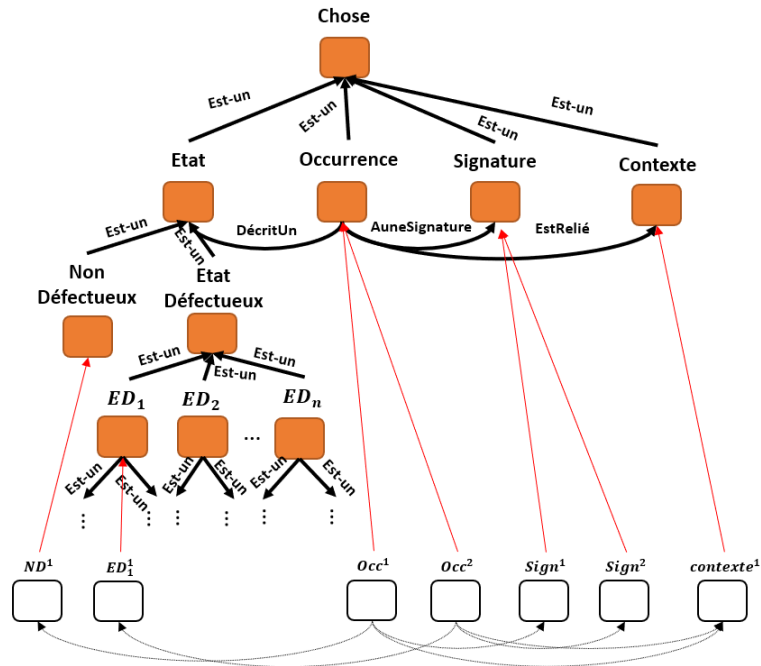


FIGURE 4.1 – Ontologie pour la Détection de Défauts (ODD) : les relations taxonomiques et sémantiques entre les concepts(en orange), et des exemples d’instances(en blanc)

significatif sur la Signature d’un État d’une opération de fabrication (correcte ou présence d’un défaut). Ce concept permettra ainsi de gérer de manière efficace les signatures des états d’opérations de fabrication suivant les contextes. Les signatures peuvent prendre différentes formes selon l’application étudiée. Par exemple, la courbe de l’énergie électrique consommée durant une opération de perçage peut être prise comme une Signature de l’État de l’opération. En outre, le concept Occurrence donne l’information quant à la répétabilité d’un État d’opération de fabrication dans un contexte particulier. Cette dernière peut être exploitée pour faire définir des indicateurs de performance pour les industriels. Enfin, il convient de noter que les attributs des classes Contexte et Signature peuvent être adaptés, étendus et modifiés selon le cas étudié.

Trois relations sémantiques qui ont le même domaine (i.e. Occurrence) sont définies. "AuneSignature" et "DécritUn" indiquent respectivement que chaque Occurrence est liée à une Signature et à un État de fabrication spécifique. "EstRelié" relie le couple <État, Signature> à son contexte; c’est-à-dire que l’ODD associe chaque Signature d’un État d’opération de fabrication à son contexte. Enfin, le nombre d’occurrences d’opération de fabrication qui partagent la même Signature (c’est-à-dire le même État d’opération de fabrication) dans un Contexte donné est stocké dans l’instance du concept Occurrence.

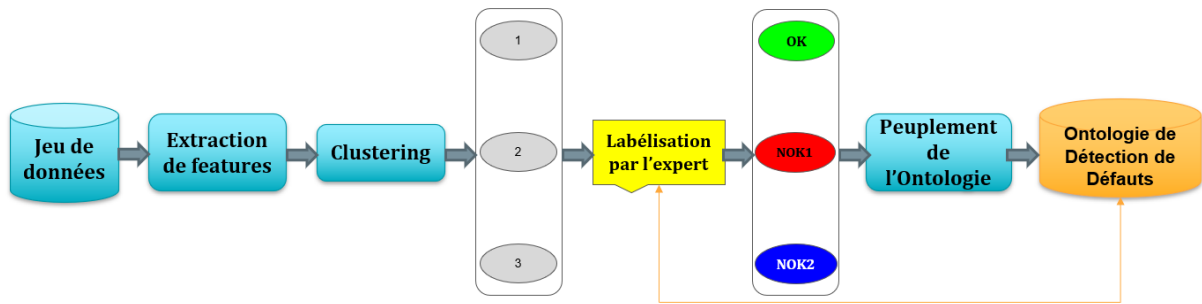


FIGURE 4.2 – Le peuplement de l'ODD à partir d'un jeu de données initial

Pour résumer, l'ODD proposée permet donc d'effectuer une surveillance process sur plusieurs contextes simultanément. Afin d'effectuer la surveillance en temps réel dans un contexte donné, l'ODD doit être peuplée d'instances des classes correctes ou défectueuses. Par conséquent, la solution proposée est composée de deux phases : un peuplement initial décrit dans la section 4.1.2 et une étape d'enrichissement décrite dans la section 4.1.3.

#### 4.1.2 Peuplement de l'ODD

**Objectifs et aperçu général** L'objectif de la première phase est d'alimenter l'ontologie en utilisant un jeu de données initial, cf. figure 4.2. En effet, initialement, elle ne contient pas les instances qui vont permettre le monitoring de production industrielle. Le jeu de données est un historique récent des signaux de capteurs de surveillance d'opérations de fabrication dans un certain Contexte. La phase d'initialisation a deux objectifs. Le premier consiste à identifier l'État d'opération de fabrication effectuée correctement, ainsi que la liste des potentiels défauts présents dans le jeu de données. Le deuxième objectif vise à identifier les Signatures de la classe correcte (i.e. "NonDéfectueux") ainsi que de chaque défaut détecté sur le Contexte dans lequel l'initialisation a été effectuée.

Pour ce faire, nous avons proposé d'utiliser une technique d'apprentissage automatique non-supervisée sur l'ensemble des données d'initialisation. À l'issue de l'apprentissage, des clusters seront appris et illustrés à un expert du domaine afin d'être labélisés. Enfin, des instances des concepts de l'ODD seront créés afin d'enregistrer les résultats de l'apprentissage (clusters et labels).

Dans ce travail, nous proposons de modéliser une signature par une distribution gaussienne dans l'espace des caractéristiques qui sont extraites des signaux des capteurs. Le jeu de données est traité de manière non-supervisée car dans un contexte de production



TABLE 4.2 – Ontologie pour la détection de défauts : définition des attributs

Classe	Attributs	Définition
Contexte	MachineId	l'Id de la machine utilisée
	StratégieId	l'Id de la stratégie process
	ManufId	l'Id du site de fabrication
Signature	...	...
	Moyenne	vecteur moyenne de la gaussienne
	Covariance	matrice de covariance de la gaussienne
Occurrence	...	...
	$N_i$	nombre d'occurrence du $i^{\text{ème}}$ couple (Signature-État)
État	État	l'état de l'opération de fabrication (OK-Défaut)
	Label	le label de l'état (étiquette)

industrielle intensive, les jeux de données ne peuvent généralement pas être labélisés. Le jeu de données peut contenir des observations de plusieurs signatures de plusieurs états non-étiquetés. Par conséquent, la distribution entière des données d'initialisation est modélisée par un modèle de mélange de gaussiennes (GMM, chapitre 2).

Afin de saisir les informations acquises dans l'ODD, nous proposons une description complète des classes de l'ODD et de leurs attributs dans le tableau 4.2. Comme évoqué précédemment, les attributs du concept Contexte dépendent de l'application étudiée. Les attributs du concept Signature résument les paramètres de chaque gaussienne apprise par le GMM. Le poids de chaque gaussienne est quant à lui enregistré à travers l'attribut  $N_i$  de l'Occurrence. En effet, comme ce dernier représente le nombre d'occurrence du  $i^{\text{ème}}$  couple (Signature-État), le poids de chaque gaussienne peuvent être ré-obtenus en calculant la fraction de chaque  $N_i$  par la somme de ces derniers qui proviennent des instances d'Occurrence sur un même Contexte. Enfin, les labels fournis par l'expert sont enregistrés dans l'attribut du concept État.

**Processus du peuplement de l'ODD** Le peuplement initial de l'ODD consiste dans un premier temps à extraire des caractéristiques significatives des signaux présents dans le jeu de données initial. Ces dernières sont supposées être capable d'identifier et de distinguer tous les défauts potentiels qui peuvent apparaître dans le processus de fabrication. Ensuite, un apprentissage non-supervisé d'un GMM par l'algorithme EM (section 2.1.3) est réalisé sur l'espace des caractéristiques extraites. Le nombre optimal de composantes du mélange est obtenu par la technique habituelle de sélection de modèles : BIC (sec-

tion 2.1.5).

Chaque gaussienne apprise est supposée correspondre à un type particulier d'opération de fabrication : défectueuse ou correcte. De ce fait, les résultats du clustering sont montrés à un expert en fabrication par le biais d'une interface homme-machine. Sur la base de ses connaissances empiriques, l'expert interprète les résultats et détermine le type de défaut (État) correspondant à chaque composante (la Signature), c'est-à-dire qu'il lui associe une étiquette (son État). Cette étape de labélisation des clusters est bien plus pertinente et efficace, que de demander à l'expert de labéliser des milliers de courbes de signaux d'opération de fabrication. En outre, cette labélisation permet dès lors d'effectuer de la classification avec le modèle appris, pour les productions à venir.

Enfin, le peuplement de l'ontologie commence par la génération de l'instance du Contexte dans lequel le GMM a été entraîné. Ensuite, les gaussiennes apprises et étiquetées sont sémantiquement liées avec les instances des classes État, Signature, Occurrence. Les étiquettes de défauts sont stockées comme des instances de la classe État. Les gaussiennes sont stockées comme des instances de la classe Signature. L'association entre les composantes gaussiennes et leurs étiquettes (c'est-à-dire les États) est assurée par le concept d'Occurrence. Enfin, les poids des composantes gaussiennes sont stockés à travers leur instance d'Occurrence  $N_i$ . Ainsi, chaque signature étiquetée apprise par le modèle GMM, est liée à un Contexte dans lequel elle a été apprise.

Par ailleurs, il convient de noter que dans le cas où certaines instances d'État ont déjà été créées dans l'ontologie par la connaissance métier ou par des apprentissages antécédents, l'expert peut visualiser l'arborescence du concept État de l'ontologie et choisir l'étiquette appropriée pour chaque composante lors de la labélisation. De plus, dans le cas où aucune des étiquettes existantes ne correspondrait à un défaut connu, l'expert peut ajouter une nouvelle étiquette du nouveau défaut qui a été découvert. Il est ajouté à l'ontologie comme un nouveau sous-concept du concept EtatDéfectueux, ce qui enrichit ainsi la structure de l'ontologie.

Par exemple, considérons une usine de production où un contexte est décrit par un croisement d'une consigne de production et de machine. Deux attributs du concept Contexte (i.e. ConsigneId, MachineId) seront alors créés. Ensuite, un apprentissage d'un GMM réalisé sur un contexte, où 1000 opérations de fabrication ont été réalisées, a identifié trois gaussiennes (deux labélisées NonDéfectueux, une labélisée  $ED_2$  par l'expert). 200 et 300 observations considérées "correctes" ont été affectées respectivement aux deux premières

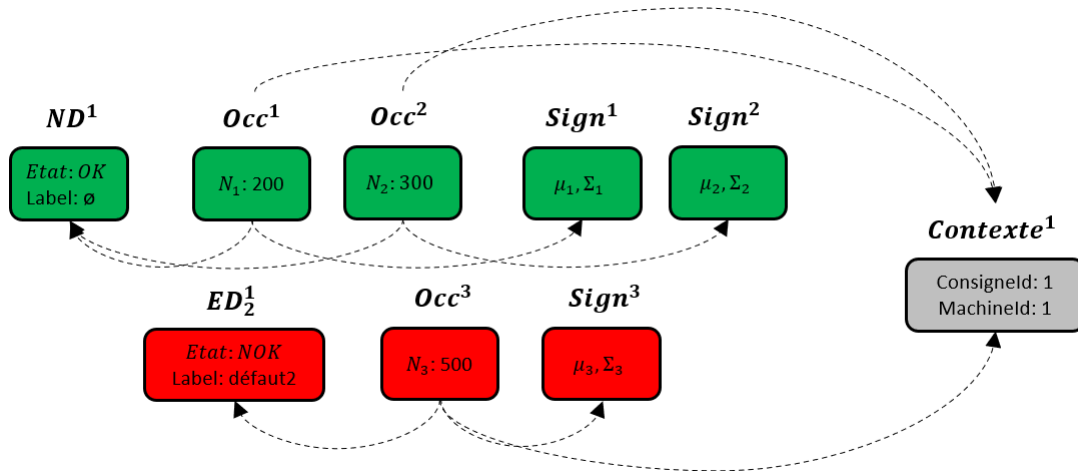


FIGURE 4.3 – Exemple d’instances créées suite à un peuplement de l’ODD sur un contexte, où 1000 opérations de fabrication ont été réalisées.

gaussiennes. Les 500 observations restantes considérées "defectueuses" ont été affectées à la dernière gaussienne. De ce fait, trois instances d’Occurrence et de Signatures seront alors créées et un lien sera fait entre chaque paire d’instance Occurrence-Signature par la relation sémantique "AuneSignature". Les attributs de chaque instance enregistrent les paramètres de chaque gaussienne apprise (vecteur moyenne et matrice de covariance dans Signature, nombre d’occurrence dans Occurrence). Ensuite, les deux premières instances d’Occurrence seront liées à l’instance du concept NonDéfectueux (nous supposons que les 2 premières gaussiennes sont labélisées NonDéfectueux), et la dernière sera liée à l’instance du concept  $ED_2$  à travers la relation "DécritUn". Enfin toutes les instances du concept Occurrence seront liées à l’instance du concept Contexte dans lequel le GMM a été appris à travers la relation "EstRelié". Les instances résultantes avec leurs attributs respectifs sont illustrées sur la figure 4.3

### 4.1.3 Enrichissement de l’ODD

Le peuplement initial de l’ODD est constitué de modèles de mélanges de gaussiennes labélisées qui permettent donc la détection et la classification de défauts par la surveillance en temps réel de la fabrication, grâce à la liste initiale de défauts qui ont été découverts et de leurs Signatures qui ont été apprises.

Nous proposons maintenant le cœur de notre architecture, dédié à la découverte itérative de connaissances à partir de nouvelles données qui peuvent être fournies dans des

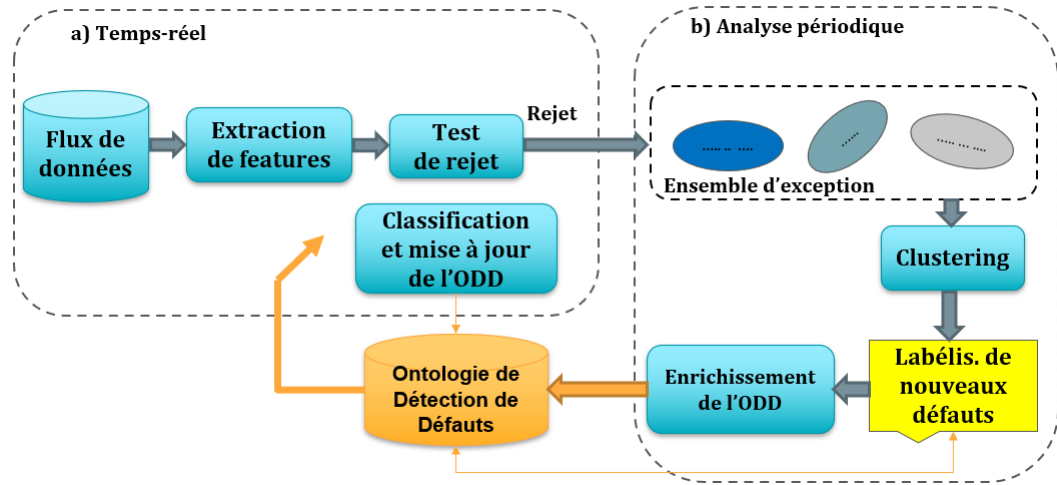


FIGURE 4.4 – Enrichissement itératif de l'ODD par une classification avec rejet sur la surveillance en continue du flux de données (a) et un mécanisme d'inspection des rejets (b)

flux de données de production industrielle. Ce processus, décrit dans la Figure 4.4, est divisée en deux parties.

La première partie (Figure 4.4.a) utilise l'ontologie déjà initialisée pour classifier les données entrantes en temps réel en utilisant ses instances de Signature et d'État existantes. Cette classification avec rejet vise à reconnaître les classes correctes ou de défauts qui sont présentes dans les données entrantes et à rejeter les données inconnues.

La seconde partie est un mécanisme d'inspection des données rejetées (Figure 4.4.b), qui est périodiquement déclenché par l'expert afin de découvrir d'éventuelles nouvelles classes de défauts inconnus, ou de nouvelles signatures de défauts existants. Ces nouveaux défauts (ou défauts mis à jour) enrichissent l'ontologie initiale. Ce processus permet donc de découvrir et d'intégrer itérativement de nouveaux défauts (ou d'en renforcer d'autres) et rend la solution plus facile à mettre en œuvre dans un environnement industriel.

La première étape consiste à vérifier que l'observation entrante appartient à un Contexte de fabrication dans lequel des Signatures ont été apprises, i.e. un GMM a été entraîné. Ensuite, les caractéristiques sont extraites des signaux mesurés et un test de rejet est effectué. Ce dernier repose sur une formule modifiée de la section 2.25 que nous avons proposé pour vérifier si une nouvelle observation appartient ou non à une Signature connue. Cette modification consiste à pondérer les seuils locaux formulés dans (2.25) par leurs poids (i.e.  $\pi_k$ ) afin de former un seuil de rejet global. Ainsi, dans notre proposition du critère de

rejet, l'avantage est que ce dernier prend en compte les poids de chacune des gaussiennes. Formellement, une observation  $x$  est rejetée si seulement si sa vraisemblance est inférieure au seuil de rejet global :

$$f(x) < \tau \sum_{k=1}^K \frac{\pi_k}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \quad (4.1)$$

Rappelons que le paramètre  $\tau$  est le risque de première espèce et que  $K$  est le nombre total de gaussiennes d'un GMM (section 2.2.1).

Si l'observation n'est pas rejetée, elle sera assignée à la classe de Signature la plus probable par l'équation 2.9, et l'attribut de l'Occurrence  $N_i$  qui est lié à l'instance de la Signature de cette classe est incrémenté dans l'ontologie. En cas de rejet, elle est affectée à l'ensemble d'exceptions.

L'analyse périodique de l'émergence possible de défaut inconnu, en explorant cet ensemble d'exceptions, est ensuite proposée. A cette étape, la même méthode de clustering par GMM et BIC décrite dans la section 4.1.2 est appliquée à l'ensemble des exceptions et une visualisation des nouveaux clusters potentiels est proposée à l'expert. Sur la base de ces résultats, l'expert interprète le type de défauts découverts. Si un nouveau défaut est confirmé, l'ontologie peut être enrichie. Les nouvelles étiquettes fournies par l'expert sont ajoutées en tant que nouveaux sous-concepts (et leurs instances) du EtatDéfectueux, qui enrichissent la structure du EtatDéfectueux dans l'ontologie. Ensuite, ces nouvelles instances sont sémantiquement liées à leurs instances de Signatures et d'Occurrence, également ajoutées. Enfin, ces signatures sont sémantiquement liées à l'instance du Contexte traité.

Par exemple, en continuant sur l'exemple présenté à la fin de la section 4.1.2, supposons qu'un apprentissage a été réalisé sur 300 individus d'exception. Suite à cet apprentissage, un GMM avec deux gaussiennes labélisées  $ED_5$  par l'expert a été appris. 100 observations ont été affectées à la première gaussienne, et 200 observations à la seconde. La mise à jour de l'ODD consiste alors à créer deux instances de Signature et d'Occurrence en enregistrant les vecteurs moyenne, matrices de covariance et poids dans leurs attributs respectifs. Ensuite, ces instances seront sémantiquement liées à l'instance du concept  $ED_5$ , également ajoutée. Enfin, les instances d'Occurrence seront liées à l'instance du Contexte dans lequel proviennent les individus d'exceptions utilisés dans l'apprentissage (*Contexte*<sup>1</sup> dans notre exemple). Les instances résultantes avec leurs attributs respectifs sont illustrées sur la figure 4.5.

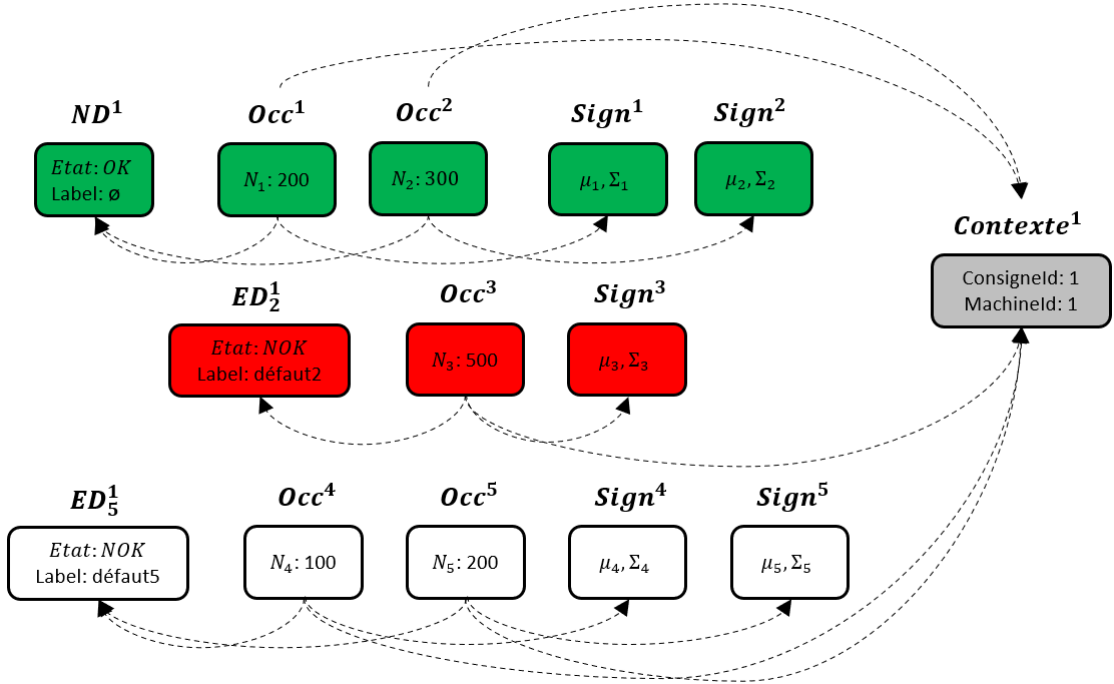


FIGURE 4.5 – Exemple d’instances créées suite à un enrichissement de l’ODD sur le même contexte pris lors du peuplement (figure 4.3), avec 300 opérations de fabrication enregistrées dans le jeu de données d’exceptions.

## 4.2 Expérimentations

Dans cette partie, nous avons testé l’efficacité de notre architecture sur plusieurs jeux de données. À cet effet, deux séries d’expériences ont été menées. Dans la section 4.2.2, nous avons étudié un comportement simple où une seule nouvelle classe de défauts apparaît à chaque itération de l’étape d’enrichissement de l’ODD, afin d’étudier la capacité à détecter de nouveaux défauts. Dans la section 4.2.3, afin d’étudier la robustesse de notre architecture, un schéma expérimental plus réaliste est proposé où des observations de plusieurs classes sont observées lors des itérations, et permettent un enrichissement progressif de l’ontologie.

### 4.2.1 Protocole Expérimental

Pour les expériences, 7 jeux de données UCI [50] ont été utilisés. Ces jeux de données sont liés à des tâches de détection de défauts où plusieurs classes décrivent des défauts ou différentes conditions de fonctionnement. Les caractéristiques sont extraites des signaux

mesurés par différents capteurs en fonction de l'application (par exemple : pression, température, ...).

Une description de ces jeux de données est fournie dans le tableau 4.3 en terme de nombre de : classes, caractéristiques et d'instances par classe. Une sélection de variables a été effectuée pour le premier jeu de données selon la méthode proposée par Tobias et al [68], ce qui a permis de réduire le nombre de caractéristiques à 25. Pour les jeux de données hydro, chaque observation (instance) est décrite par 17 séries temporelles mesurées à partir des différents capteurs physiques (ex : capteur de pression). Pour extraire les caractéristiques, nous retenons la valeur maximale de chaque série temporelle, ce qui nous donne 17 caractéristiques pour chaque observation. Les deux derniers jeux de données (Gas Sensor Drift, Steel Faults) ont été laissés inchangés. Enfin, une normalisation standard dans l'espace des caractéristiques a été appliquée à tous les jeux de données.

Il convient de souligner que ces jeux de données sont utilisés de manière non-supervisée par notre architecture. L'information fournie par la valeur de la classe n'est utilisée que pour simuler l'étape de labélisation par l'expert proposée dans notre architecture. Cet étiquetage est effectué en associant chaque Signature découverte à l'étiquette de la classe majoritaire dans ce cluster. Dans le cas où la différence de probabilités des deux premières classes majoritaires (équation 2.34) est inférieure à un seuil donné (équation 2.31 avec  $\Theta_a$  fixé à 0.3 sur la base de l'expérimentation), la gaussienne sera rejeté par ambiguïté. En effet, nous estimons alors que, de même, l'expert observant un mélange de deux défauts dans un même cluster, il n'aurait pas pu déterminer clairement que ce cluster relève de tel défaut. En cas de telle ambiguïté, tout le jeu de données associé à ce cluster sera placé dans l'ensemble des exceptions. Ainsi, l'étiquetage est effectué automatiquement dans ces expériences.

L'architecture proposée a été implémentée en Python avec sklearn pour l'apprentissage de modèles et l'option de rejet. Le nombre maximal de cluster  $K_{max}$  a été fixé à 50 et  $\tau$  à 1% pour le test de rejet (équation 4.1). Le type de paramètres de la matrice de covariance à utiliser a été fixé à "full". Par ailleurs, Owlready2 [90] a été utilisé pour l'implémentation de l'ontologie et l'interaction avec les objets Python.

TABLE 4.3 – Description des jeux de données

Jeux de données	$n_{classes}$	$n_{features}$	$n_{instances}$	$n_{instances-par-classe}$
Sensorless Drive Diagnosis	11	49	58509	5319-...-5319
Steel Faults	7	27	1941	158-190-391-72-55-402-673
Hydro Hydro	4	17	2205	599-399-399-808
Hydro Cooler	3	17	2205	741-732-732
Hydro Leakage	3	17	2205	1221-492-492
Hydro Valve	4	17	2205	1125-360-360-360
Gas Sensor Drift	6	129	13910	2565-2926-1641-1936-3009-1833

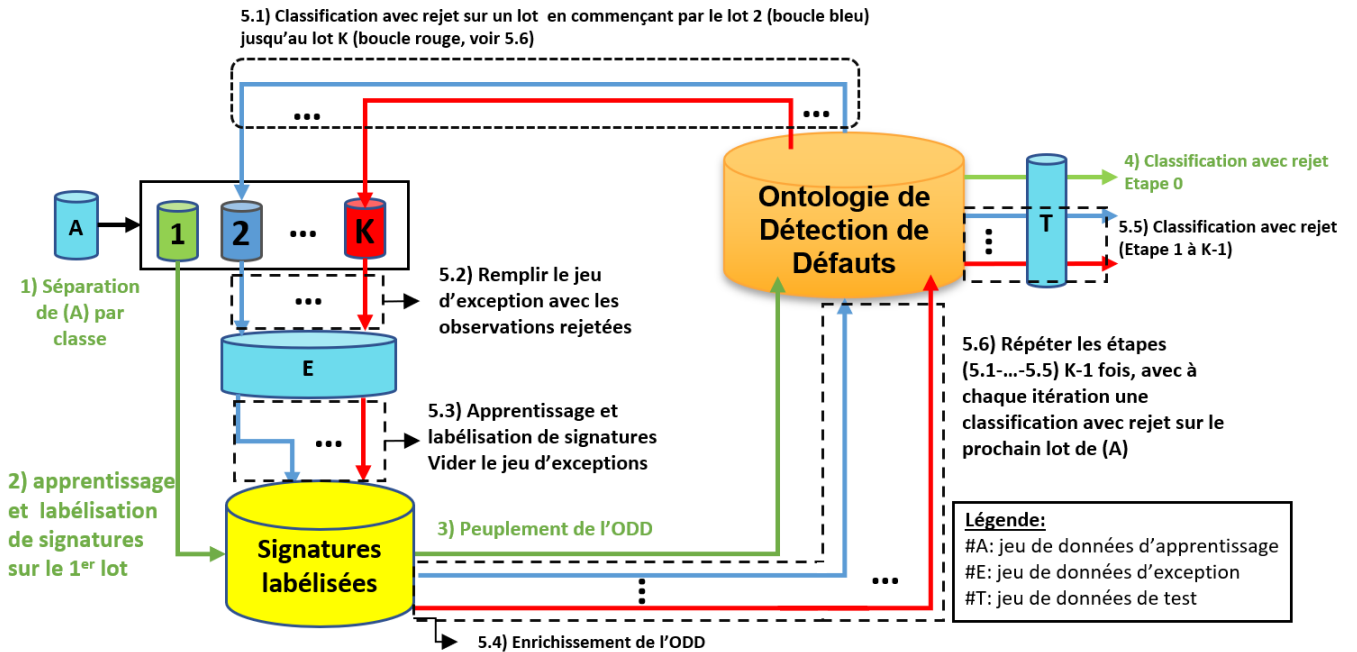


FIGURE 4.6 – Description de l'expérience de détection et apprentissage de nouveaux défauts

### 4.2.2 Détection et apprentissage de nouveaux défauts

La principale propriété de l'architecture proposée est liée à la capacité à détecter de nouveaux défauts de manière itérative.

À cette fin, chaque jeu de données a été divisé en un ensemble d'apprentissage et un ensemble de test avec une répartition aléatoire. Ensuite, l'ensemble d'apprentissage a été séparé par classe. Chaque lot contient des instances d'une seule classe. Le peuplement initial de l'ODD a été effectuée sur le premier lot (appelé étape 0 dans les figures 4.7



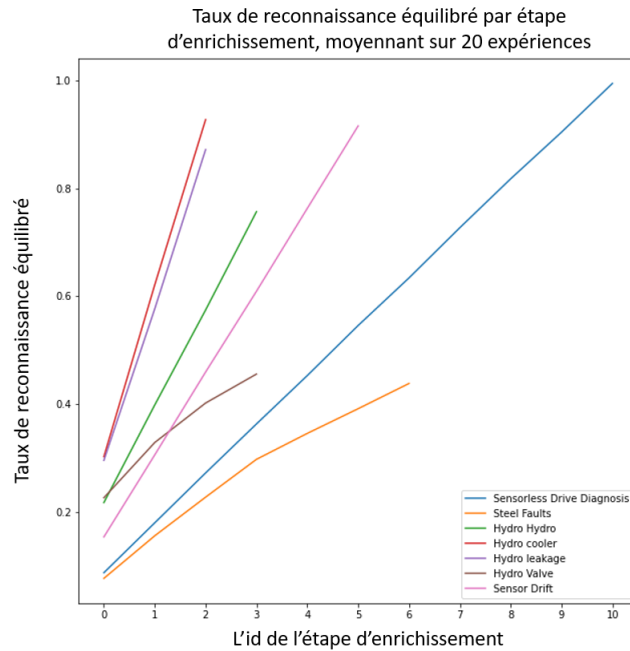


FIGURE 4.7 – Détection et apprentissage de nouveaux défauts : précision équilibrée globale du système dans le peuplement initial de l’ODD (étape 0) et pendant le processus itératif de découverte de connaissances (en moyennant sur 20 expériences différentes).

et 4.9). Ensuite, l’enrichissement itératif de l’ODD est successivement effectué sur les lots restants. Un schéma présentant l’expérience est illustré sur la figure 4.6.

Nous proposons ici de mesurer la capacité à détecter de nouveaux comportements par deux métriques. Tout d’abord, le *taux de reconnaissance par classe* qui reflète la capacité pour un nouveau défaut donné d’être rejeté par la classification en temps réel (basée sur l’ODD qui contient déjà des défauts, mais pas encore ce défaut ci). Il s’agit d’une première condition nécessaire pour que les données de cette nouvelle classe soient rejetées, puis traitées par le mécanisme d’inspection et potentiellement ajoutées à l’ontologie. Ce critère est évalué par le taux de rejet (nombre d’observations rejetées sur le nombre total d’observations) sur le jeu de données de test. Ensuite, le *taux de reconnaissance des classes* est la capacité à bien classer un défaut connu, une fois les Signatures correspondantes ajoutées dans l’ontologie. Cette capacité peut être mesurée par la précision globale (nombre de prédictions correctes sur le nombre total d’observations) estimée sur le jeu de données de test. Enfin, comme certains jeux de données sont déséquilibrés, nous proposons d’utiliser sa version équilibrée (i.e. balanced accuracy) dans la plupart des expériences [19].

Afin de limiter le biais possible lié à l’ordre d’introduction des défauts dans ce scénario

TABLE 4.4 – Détection et apprentissage de nouveaux défauts : taux de reconnaissabilité et de reconnaissance par défaut (moyenne sur 20 expériences avec un ordre d’apparition de défauts différent)

Jeux de données	défaut	taux de reconnaissabilité	taux de reconnaissance
Sensorless Drive Diagnosis	0	0.99	1.00
	1	0.99	0.99
	2	0.99	1.00
	3	1.00	1.00
	4	0.99	1.00
	5	0.99	0.99
	6	1.00	1.00
	7	0.99	0.99
	8	1.00	1.00
	9	0.99	1.00
	10	1.00	1.00
Steel Faults	0	0.78	0.48
	1	0.88	0.57
	2	0.99	0.66
	3	0.97	0.72
	4	0.78	0.42
	5	0.81	0.65
	6	0.90	0.68
Hydro Hydro	0	1.00	0.80
	1	0.99	0.87
	2	0.99	0.93
	3	1.00	0.94
Hydro Cooler	0	1.00	0.98
	1	1.00	0.98
	2	1.00	0.89
Hydro Leakage	0	1.00	0.87
	1	1.00	0.93
	2	1.00	0.92
Hydro Valve	0	0.93	0.75
	1	0.82	0.69
	2	0.81	0.63
	3	0.82	0.77
Gas Sensor Drift	0	1.00	0.91
	1	0.99	0.83
	2	1.00	0.90
	3	0.82	0.93
	4	1.00	0.97
	5	0.99	0.97

d’apprentissage, ces métriques sont moyennées sur 20 expériences avec différents ordres d’apparition de défauts.

Le tableau 4.4 présente les résultats des taux de reconnaissabilité et de de reconnais-

sance des classes pour chaque défaut présent dans les 7 jeux de données. Ici, le taux de reconnaissance de classe non-équilibré (i.e, unbalanced accuracy) est utilisé afin de le comparer aux autres travaux qui ont utilisé la même métrique. Nous pouvons constater que notre architecture est capable de rejeter les nouveaux défauts qui n'ont pas été vus précédemment, indépendamment de leur ordre d'introduction dans ce scénario. Le taux de reconnaissabilité est très élevé, ce qui signifie que les défauts inconnus sont principalement rejetés pendant la classification en temps réel, et pourront ensuite être traités par le mécanisme d'inspection suivant. La dernière colonne présente le taux de reconnaissance des classes, c'est-à-dire la façon dont ce défaut est reconnu une fois que les Signatures correspondantes ont été apprises et ajoutées dans l'ontologie par le mécanisme d'inspection. Ce taux est globalement bon, hormis pour deux jeux de données où les classes sont plus confuses et difficiles à distinguer. Les valeurs présentées ici sont similaires au taux de classification que l'on peut observer pour de tels jeux de données dans la littérature : [73], [154], [167]. Enfin, nous pouvons voir sur la figure 4.7 que la précision globale équilibrée augmente de façon monotone au cours des itérations pour tous les jeux de données, ce qui signifie que l'enrichissement de l'ontologie améliore progressivement la précision globale malgré le déséquilibre entre classes dans certains jeux de données.

### 4.2.3 Robustesse

Dans l'expérience précédente, un premier scénario d'apprentissage a été proposé où une seule nouvelle classe était observée à chaque itération pendant l'enrichissement itératif de l'ODD. Nous proposons ici d'étudier un scénario d'apprentissage plus complexe, en considérant un schéma plus réaliste où des instances de plusieurs classes sont observés pendant une itération.

Comme précédemment, chaque jeu de données a été divisé en jeux d'entraînement et de test. Puis, l'ensemble d'apprentissage a été mélangé de façon aléatoire afin que les instances de toutes les classes soient distribuées de façon uniforme. Ensuite, le peuplement de l'ODD a été effectuée sur les premières  $n_{init}$  instances. Enfin, l'enrichissement itératif de l'ODD a été effectué sur les instances restantes, en lançant le mécanisme d'inspection périodique lorsqu'un nombre donné d'instances rejetées  $n_{rejet}$  est atteint. Comme précédemment, cette procédure a été répétée 20 fois en mélangeant à chaque fois l'ensemble d'apprentissage.

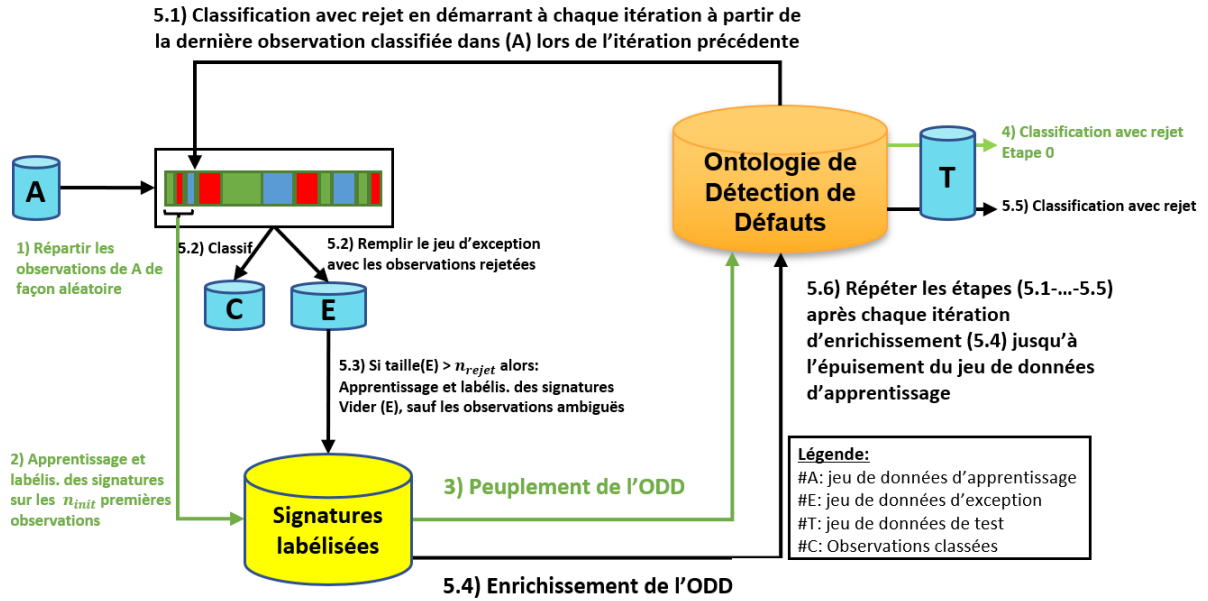


FIGURE 4.8 – Description de l'expérience de la robustesse

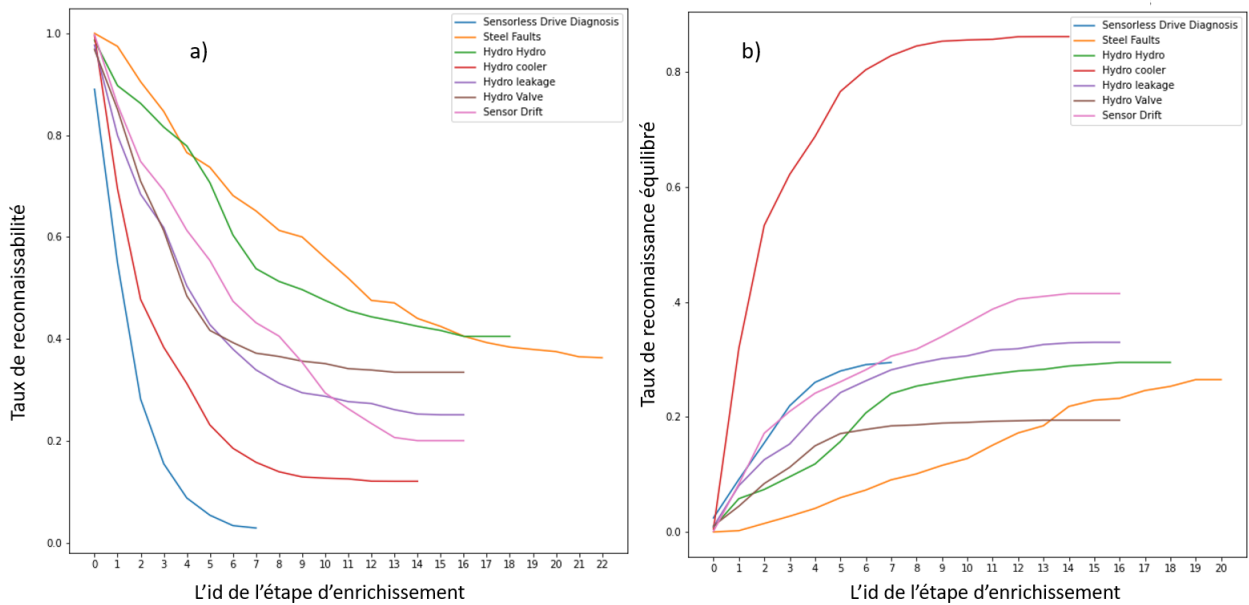


FIGURE 4.9 – Robustesse : a) taux de reconnaissabilité, b) taux de reconnaissance équilibré, dans Le peuplement initial de l'ODD (step 0) et pendant le processus itératif de découverte de connaissances, (en moyennant sur 20 expériences différentes).

$n_{init}$  et  $n_{rejet}$  ont été fixés respectivement à 2% et 4% de la taille de l'ensemble d'apprentissage.

Nous proposons de mesurer la robustesse de notre architecture à l'aide de deux métriques. Premièrement, le *taux de reconnaissabilité* est encore défini comme le taux de rejet mesuré ici à chaque étape de l'enrichissement de l'ontologie sur le jeu de données de test. La métrique devrait diminuer de 1 à 0 au cours de notre enrichissement itératif de l'ODD, puisque les signatures correspondant à chaque défaut sont progressivement ajoutées. Dans le même temps, le *taux de reconnaissance équilibré* mesure la capacité à bien classer les défauts, une fois que leurs signatures correspondantes ont été ajoutées dans l'ontologie. Cette métrique devrait passer de 0 à 1 si les défauts sont bien reconnus dans le jeu de données de test. Comme précédemment, la moyenne de ces métriques est calculée sur 20 expériences avec différents mélanges aléatoires de données.

Les résultats sont présentés dans les deux figures 4.9 (a. et b.). Comme prévu, nous pouvons observer, pour tous les jeux de données, que l'étape de classification en temps réel rejette de moins en moins d'instances à mesure que l'ontologie est enrichie. De plus, le taux de reconnaissance augmente au cours de ces étapes d'enrichissement, ce qui signifie que les défauts et les signatures ajoutés progressivement dans l'ontologie sont utiles pour détecter les défauts correspondants dans les étapes suivantes.

Cela montre que notre architecture est robuste contre la distribution aléatoire des défauts dans l'ensemble d'apprentissage, ce qui la rend robuste dans un contexte industriel où les défauts peuvent survenir à tout moment.

### 4.3 Conclusion

Dans ce chapitre, nous avons proposé une architecture hybride (ML-KM) dédiée à la découverte itérative de connaissances pour la détection de défauts dans les systèmes de fabrication. La première contribution de cette partie est une ontologie générique de détection de défauts (ODD) par surveillance de la fabrication. La deuxième et troisième contribution correspond à la proposition de mécanisme d'apprentissage automatique non-supervisé, permettant de détecter des défauts en alimentant l'ontologie dans un premier temps (peuplement), puis de l'enrichir de manière itérative lorsque de nouveaux défauts, potentiellement inconnus, sont découvertes dans le flux de données en temps réel. Ce mécanisme permet ainsi de découvrir de nouvelles connaissances par le biais de l'apprentissage automatique qui enrichie l'ontologie progressivement. Un dernier apport est la proposition d'une labélisation des clusters par l'expert ; qui est bien plus efficace qu'un laborieux

étiquetage de milliers d'essais. De plus, les connaissances formalisées par l'expert seront capturées et réutilisées par l'ontologie.

Afin de valider l'approche, des expériences ont été menées sur 7 jeux de données de l'UCI liés à la fabrication ou aux systèmes industriels. Elles ont démontré la capacité de notre architecture à détecter de nouveaux défauts de manière itérative (section 4.2.2), et à ajouter des informations pertinentes dans l'ontologie permettant une meilleure reconnaissance des défauts dans les étapes suivantes, de manière robuste (section 4.2.3).

Dans notre architecture, le lien entre les techniques d'apprentissage automatique et l'ingénierie des connaissances est établi à travers une ontologie gérant plusieurs GMM labélisés. La labélisation des clusters est effectuée de façon intelligente par l'expert, ce qui permet d'éliminer la charge de labéliser manuellement toutes les observations. De plus, comme dans tout processus d'élicitation en ingénierie des connaissances, cette interaction avec l'expert, guidée par la fouille de donnée, leur permet de formaliser et modéliser des connaissances tacites.

Ce chapitre se focalisait sur la détection et la gestion de l'apparition de nouveau défaut, dans un contexte donné. Dans le chapitre suivant, le concept de Contexte décrit dans notre ontologie sera pris en compte pour étudier comment la détection de défauts peut être effectuée sur des données en temps réel provenant d'un nouveau contexte expérimental, en ré-utilisant la connaissance des signatures de défauts apprises dans des contextes similaires. La généralisation des connaissances ajoutées progressivement dans l'ontologie sera également étudiée. Par exemple, peut-on fusionner plusieurs signatures correspondant à un même défaut en une signature unique? Peut-on généraliser une signature identifiée pour un défaut donné à d'autres contextes? Ces questions sont liées aux domaines de l'apprentissage par transfert et de l'adaptation au domaine, appliqués ici pour induire de nouvelles connaissances dans notre ontologie ODD, et augmenter les performances du système de détection de défauts.



# GÉNÉRALISATION ET EXPLOITATION DES CONNAISSANCES

---

Dans ce chapitre, la prochaine étape de notre architecture dédiée à la découverte itérative de connaissances pour la détection de défauts dans les systèmes de fabrication sera présentée. Cette étape consiste à exploiter les connaissances issues de contextes similaires afin d'effectuer la détection de défauts sur des données provenant d'un nouveau contexte. À cet effet, des mécanismes de généralisation de connaissances ont été formalisés et testés sur des données de l'UCI. Ce chapitre est organisé comme suit : la formalisation du problème est présentée dans la section 5.1, puis une solution est proposée dans la section 5.2. La section 5.3 présente les expériences sur les jeux de données de l'UCI. Enfin, les conclusions et perspectives sont tirées dans la section 5.4.

## 5.1 Formalisation du problème

Nous avons démontré dans le chapitre 4 que notre architecture est capable de détecter de nouveaux défauts de manière itérative et robuste, en enrichissant progressivement l'ontologie. La prochaine étape de l'évolution de notre architecture est maintenant proposée. Cette évolution sert à satisfaire les trois besoins suivants que nous avons identifiés.

Tout d'abord, comme des connaissances sont ajoutées dans l'ontologie au fil du temps, les performances de notre système pourraient être diminuées à terme. En effet, l'ajout continu de signatures peut saturer l'ontologie ce qui augmentera le temps de classification. De plus, il est probable qu'il y aura une accumulation de redondances (sur-apprentissage) et/ou d'obsolescences (sous-apprentissage) de signatures. Ainsi, il est nécessaire de concevoir un mécanisme permettant de réguler le nombre de signatures dans le temps.



Ensuite, rappelons que l'objectif de notre système est d'effectuer la surveillance process dans plusieurs contextes industriels. Or, il est probable qu'il y aurait peu de données dans certains contextes pour un peuplement de l'ontologie (chapitre 4.1.2), ce qui empêcherait d'effectuer un peuplement de l'ontologie et de la classification sur ces contextes. Ainsi, ces contextes resteront inconnus par l'ontologie, car aucune signature n'aura été apprise. Par conséquent, il faudrait définir un mécanisme permettant d'exploiter les connaissances déjà apprises dans des contextes similaires afin de classifier les observations dans ces nouveaux contextes.

Enfin, comme certains défauts peuvent être rencontrés sur un nombre limité de contextes, il est nécessaire de transférer ces connaissances vers d'autres contextes où leurs probabilités d'apparitions est très faible. Cela permettra donc de détecter ces très rares défauts dans ces contextes.

Cet aspect d'explorer les connaissances capturées dans l'ontologie à travers les techniques de ML permettra, à son tour, de définir des connaissances plus génériques permettant d'améliorer les techniques de ML utilisées. Cette amélioration consistera à augmenter la qualité de la classification dans des contextes connus (régulation des signatures). De plus, elle permettra de classifier des observations issues de contextes inconnus, où peu de données existent pour effectuer une surveillance.

## 5.2 Description de la solution

Pour répondre aux besoins exprimés dans la section 5.1, nous proposons dans un premier temps de faire évoluer la structure de l'ontologie décrite dans la section 4.1. Ensuite, nous proposons dans un second temps deux mécanismes d'exploitation des connaissances : la réduction intra-contexte et la généralisation intercontextes. Enfin, nous décrivons l'algorithme utilisé dans l'exploitation des connaissances.

### 5.2.1 Évolution de la structure l'ontologie

Nous avons décrit dans le chapitre 4.1 qu'un contexte est défini par un seul concept général. En pratique, un contexte peut être décrit par une multitude de concepts interreliés. Par exemple, plusieurs gammes d'outils ou de machines qui sont utilisées peuvent exister, avec divers paramètres opératoires dans différents ateliers au sein d'une usine. Il

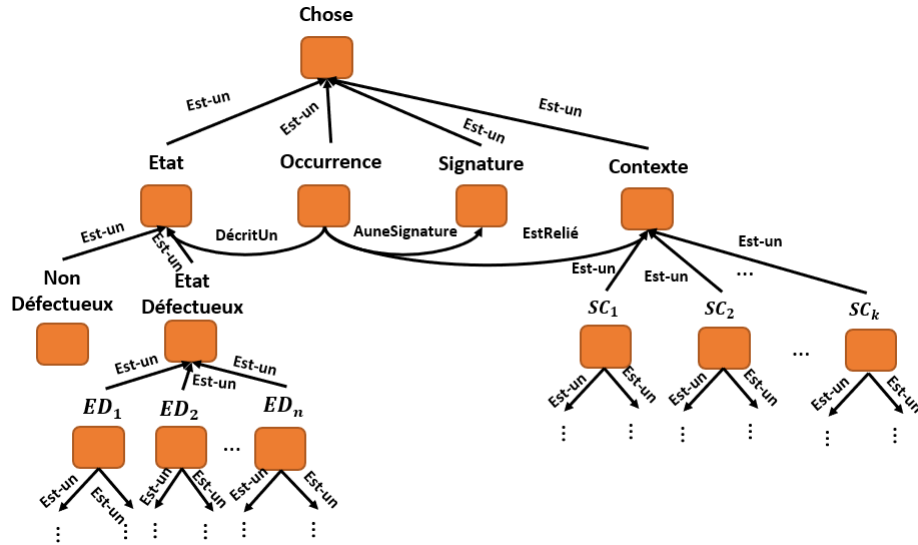


FIGURE 5.1 – Évolution de l'Ontologie pour la Détection de Défauts (ODD)

est clair que ceux-ci nous ne peuvent pas être décrits par un seul concept. De ce fait, il est nécessaire de définir explicitement des concepts formant le contexte ainsi que leurs interactions afin de gérer efficacement les connaissances apprises.

À cet effet, nous proposons une évolution de l'architecture de l'ontologie décrite par la figure 5.1. L'architecture proposée est générique et peut être adaptée à l'application étudiée. Le concept "SC" est une abréviation de "sous-contexte". Par exemple, le concept "Contexte" peut référer une gamme de machine, tandis que les sous-concepts (SC) réfèrent des machines spécifiques mais similaires.

Dans ce qui suit, un seul niveau de hiérarchie de contextes sera considéré, pour simplifier les explications. C'est à dire que la partie contexte sera décrite par le concept "Contexte" ainsi que ces sous-contextes directs ( $SC_1, SC_2, \dots, SC_k$ ). Les mécanismes décrits ci-dessous peuvent être appliqués à différents niveaux de contextes. En effet, il suffit juste d'avoir des sous-concepts héritant directement du même concept général.

### 5.2.2 Mécanisme 1 : Réduction intra-contexte

Pour ce premier mécanisme de généralisation, supposons qu'un ensemble d'instances de concepts et de relations sémantiques "A-box" est décrit comme suit :

- $SC_1^1$  Est-une instance du sous-Concept de Contexte  $SC_1$
- $ND^1$  Est-une Non Défectueux

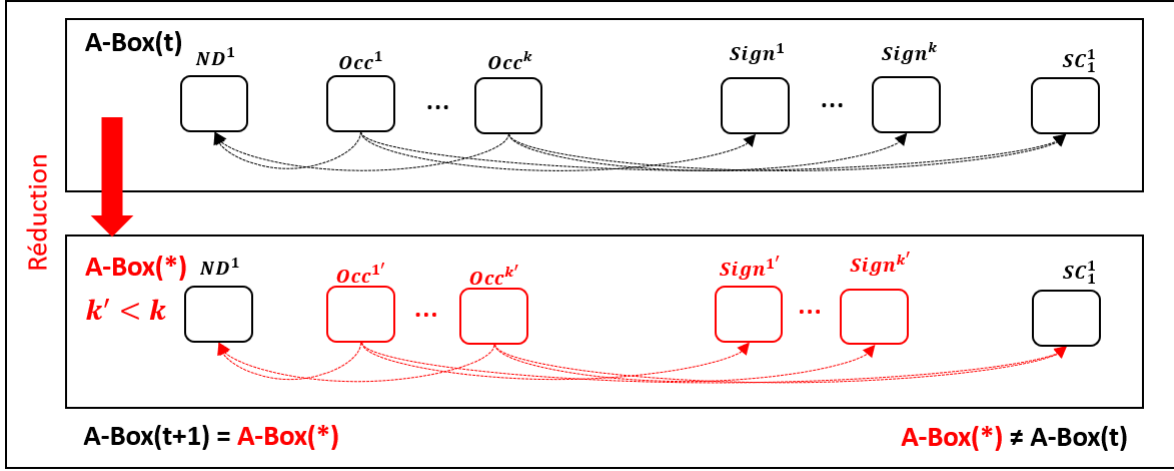


FIGURE 5.2 – Mécanisme de réduction de signatures

- $(Sign^1, Sign^2, \dots, Sign^k)$  Est-une Signature
- $(Occ^1, Occ^2, \dots, Occ^k)$  Est-une Occurrence
- $AUneSignature((Occ^1, Sign^1), (Occ^2, Sign^2), \dots, (Occ^k, Sign^k))$
- $DécritUn((Occ^1, ND^1), (Occ^2, ND^2), \dots, (Occ^k, ND^k))$
- $EstRelié((Occ^1, SC_1^1), (Occ^2, SC_1^2), \dots, (Occ^k, SC_1^k))$

Afin de simplifier la "A-box" décrite ci-dessus, nous proposons de l'écrire comme suit :

- $\{(ND^1, SC_1^1, Occ^1, Sign^1), \dots, (ND^1, SC_1^1, Occ^k, Sign^k)\}$

L'objectif est de définir un mécanisme permettant de remplacer la "A-box" décrite ci-dessus par la nouvelle "A-box" (figure 5.2) :

- $\{(ND^1, SC_1^1, Occ^{k'}, Sign^{k'}), \dots, (ND^1, SC_1^1, Occ^{k'}, Sign^{k'})\}$ , avec  $k' < k$ .

Il convient de noter que cette définition est appliquée ici à l'état correcte (i.e.  $ND^1$ ), mais qu'elle peut être appliquée pour chaque type de défaut (i.e.  $ED$ ).

Ce mécanisme permet de créer de nouvelles instances des concepts Occurrence et Signature, puis de définir leurs attributs respectifs. Concrètement, il faut définir les vecteurs moyennes  $\mu_i$  et matrices de covariances  $\Sigma_i$  des signatures associées aux nouvelles instances  $Sign^i$ , ainsi que leur nombre d'occurrence  $N_i$  des instances  $Occ^i$ , avec  $i \in \{1', \dots, k'\}$  et  $k' < k$ . La méthode d'estimation de ces nouvelles signatures réduites sera élaborée à la section 5.2.4 qui va suivre. Notons que ces nouvelles instances sont toujours liées respectivement aux mêmes instances de défauts et de sous-contexte ( $ND^1, SC_1^1$ ).

La création de ces instances traduit le fait que nous avons réduit le nombre de si-

gnatures liées à l'état  $ND^1$  et au même sous-contexte  $SC_1^1$ . Cette réduction permet de réduire les signatures obsolètes et/ou redondantes apprises au cours de la classification, tout en retenant le maximum d'information. Ainsi, ce mécanisme permettra de satisfaire le premier besoin énuméré dans la section 5.1. De plus, rappelons que ce mécanisme peut être appliqué à tout Sous-Contexte et à tout  $\acute{E}tat$ ; le couple  $(ND^1, SC_1^1)$  n'étant qu'une illustration.

Enfin, comme ce mécanisme réduit le nombre de signatures tout en retenant un maximum d'information, et que chaque signature est décrite par une gaussienne, nous pouvons affirmer que la réduction de signatures est équivalente à l'application du Gaussian Mixture Reduction (GMR) décrit dans le chapitre 2.3. Ainsi, nous allons nous baser sur des algorithmes de GMR pour proposer la réduction de signatures similaires intra-contexte.

### 5.2.3 Mécanisme 2 : Généralisation inter-contextes

Dans cette partie, supposons que l'on dispose des instances de concepts et de relations sémantiques décrites comme suit :

- $\{(ND^1, SC_1^1, Occ_1^1, Sign_1^1), \dots, (ND^1, SC_1^1, Occ_1^i, Sign_1^i)\}$ , avec  $i \in \{1, \dots, n_1\}$ ,
- ...
- $\{(ND^1, SC_k^1, Occ_k^1, Sign_k^1), \dots, (ND^1, SC_k^1, Occ_k^i, Sign_k^i)\}$ , avec  $i \in \{1, \dots, n_k\}$  et  $n = \sum_{j=1}^k n_j$ .

En effet, on considère que des signatures ont été apprises sur  $k$  sous-contextes héritant du concept "Contexte" et associées à la même classe de défaut  $ND^1$ . En d'autres termes, des signatures de la même classe de défauts sur plusieurs sous-contextes ont été apprises. Pour faciliter la compréhension, les indices sous les instances des occurrences et signatures indiquent les sous-contextes auxquels elles sont liées.

L'objectif est maintenant de proposer un mécanisme permettant d'enrichir la "A-box" avec les instances suivantes (figure 5.3) :

- $Contexte^1$  Est-un Contexte,
- $\{(ND^1, Contexte^1, Occ_C^1, Sign_C^1), \dots, (ND^1, Contexte^1, Occ_C^i, Sign_C^i)\}$ ,  
avec  $i \in \{1, \dots, m\}$  et  $m < n$ .

L'indice "C" sous les instances des occurrences et signatures indique que ces dernières sont liées à l'instance  $Contexte^1$ .

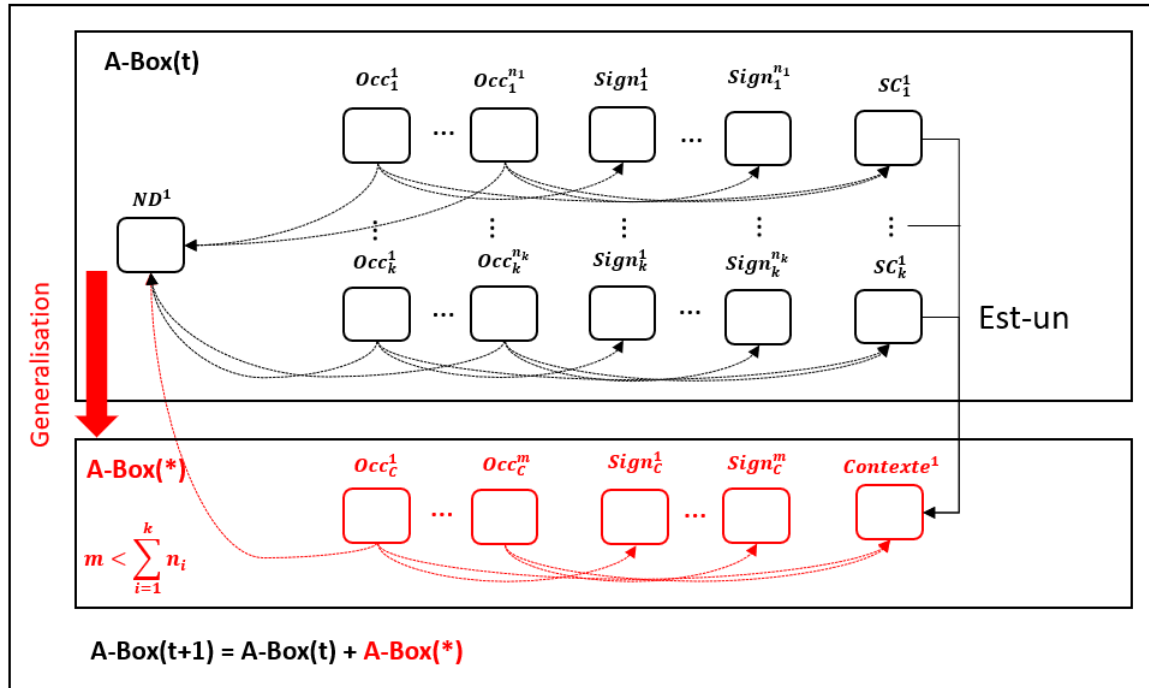


FIGURE 5.3 – Mécanisme de généralisation de signatures

Ce mécanisme permet donc de créer des signatures "généralisées" liées au concept général, instancié par "Contexte<sup>1</sup>" ici. Ces dernières seront créées à partir des signatures "spécifiques" liées à chacun des sous-concepts héritant du "Contexte<sup>1</sup>". Les signatures généralisées ont pour but d'agréger les connaissances apprises sur chacun des sous-concepts afin de les rendre indépendantes des sous-concepts. Ainsi, par analogie au premier scénario, ce mécanisme de généralisation agrégera avec réduction les signatures similaires inter-contextes et gardera les signatures différentes inchangées. Par conséquent, comme pour la réduction intra-contexte, ce mécanisme est équivalent à appliquer un GMR sur les signatures inter-contextes.

Ces signatures généralisées pourront ensuite être exploitées par tous les sous-contextes héritant du contexte général. Ainsi, la classification des observations sur des contextes inconnus pourra être initiée en se basant sur les signatures généralisées sans attendre d'avoir assez de données pour effectuer un apprentissage de signature et un peuplement de l'ontologie d'ODD (section 4.1.2).

Formellement, comme pour le premier scénario, il faudra créer dans un premier temps de nouvelles instances des concepts Occurrence et Signature, de définir leurs attributs respectifs (i.e,  $\mu_i$  et matrices de covariances  $\Sigma_i$  des instances  $Sign_c^i$  et  $N_i$  des instances

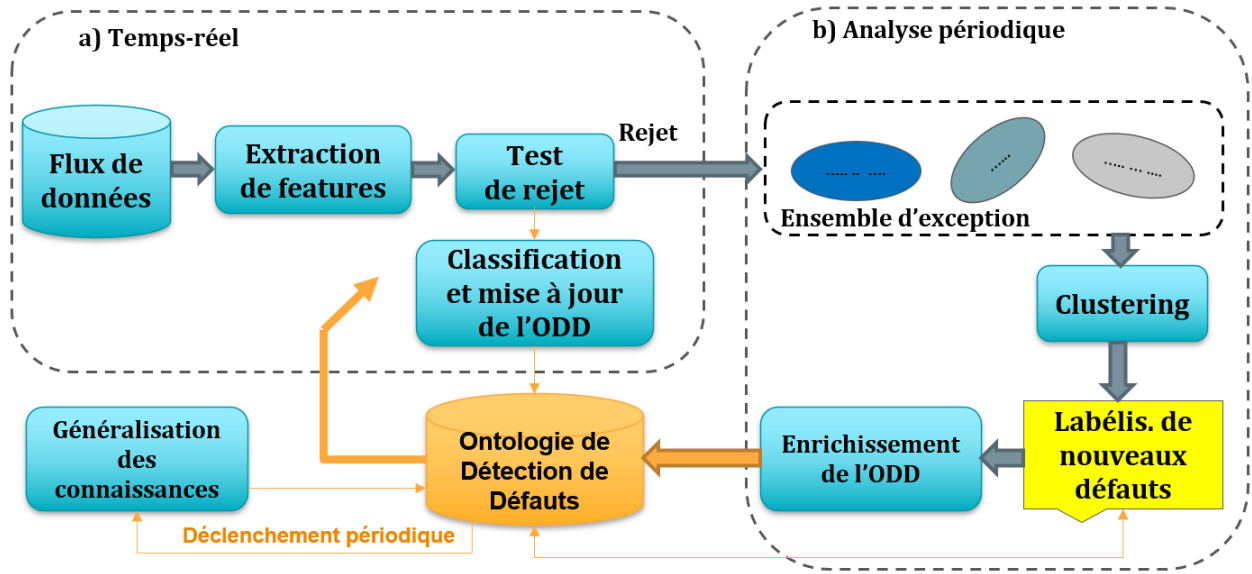


FIGURE 5.4 – Nouvelle architecture de l’enrichissement itératif de l’ODD par une classification avec rejet sur la surveillance en continue du flux de données (a), un mécanisme d’inspection des rejets (b) et une généralisation des connaissances (mécanismes 1 et 2).

$Occ_C^i$  avec  $i \in \{1, \dots, m\}$ ).

Ainsi, les deux mécanismes de généralisation de connaissances peuvent être appliqués de façon périodique, ou à la demande de l’expert, en se basant sur les connaissances capturées par l’ontologie. Par exemple, il peuvent se déclencher suite à chaque enrichissement de l’ODD avec de nouvelles connaissances apprises lors de l’inspection périodique des rejets. La nouvelle architecture de notre système de surveillance est alors illustrée par la figure 5.4.

#### 5.2.4 Algorithme de réduction de modèle gaussien

Rappelons que les mécanismes proposés précédemment consistent à réduire le nombre de signatures tout en retenant un maximum d’information. En effet, l’objectif est de fusionner les signatures similaires au sein d’un contexte, puis entre les contextes, d’éliminer celles obsolètes, pour permettre de faire hériter de ces signatures généralisées dans de nouveaux contextes. Par conséquent, comme la modélisation des signatures est représentée par des gaussiennes, le mécanisme de réduction que nous allons mettre en oeuvre est la réduction de mélanges de gaussiennes (GMR). La description algorithmique des mécanismes de généralisation décrits précédemment est illustrée par l’algorithme 3.

---

**Algorithme 3 :** mécanisme de réduction intra- ou généralisation inter-contexte, de signatures de défaut par GMR

---

**Input :** Instances de signatures et d'occurrences :  $Sign_j^i, Occ_j^i, j \in \{1, \dots, k\}, i \in \{1, \dots, n_j\}, n = \sum_{j=1}^k n_j$ . Si  $k = 1$  alors, l'algorithme effectue une réduction intra-contexte, sinon généralisation inter-contexte.

1. Extraire les attributs  $N_j^i$  de chaque instance d'occurrence  $Occ_j^i$  avec  $i \in \{1, \dots, n_j\}, j \in \{1, \dots, k\}$ .
2. Normaliser les attributs  $N_j^i$  telle que leurs somme soit égale à 1 (i.e. calcul des poids  $\pi_j^i$  du GMM englobant toutes les signatures).
3. **Réduire le nombre de signatures par GMRC** en prenant en entrée les vecteurs moyenne et matrices de covariance de chaque signature  $Sign_j^i$  ainsi que les attributs  $N_j^i$  normalisés ( $N = \sum_{ij} N_j^i$ ).
4. Création de nouvelles instances d'Occurrence et de Signatures avec un nombre  $m$  égal au nombre de signatures réduites résultantes.
5. (Optionnel) Création d'une instance de Contexte (concept général) si  $k > 1$ .
6. Affecter les vecteurs moyenne et matrices de covariance de chaque signature réduite à son instance associée.
7. Dé-normaliser (multiplier par  $\mathbf{N}$  obtenu à l'étape 3) les poids des signatures résultantes et les convenir en entier pour obtenir les attributs des nouvelles occurrences.

**Output :** Instances de Signatures et d'Occurrences :  $Sign_l^q, Occ_l^q, l = 1$  ou  $l = C, q \in \{1, \dots, m\}, m < n$ . Si  $l = 1$  alors réduction intra-contexte, sinon généralisation inter-contexte.

---

**Choix de l'algorithme de réduction** Nous avons choisi l'algorithme GMRC (Gaussian Mixture Reduction with Clustering, section 2.3.5) pour effectuer la réduction de gaussiennes. Cette opération se situe à l'étape 3 (en gras) de l'algorithme 3 des mécanismes de généralisation. Rappelons que cet algorithme effectue dans un premier temps, une réduction gloutonne sur les gaussiennes du GMM initial en appliquant l'algorithme de Runnals 1, afin de définir des gaussiennes réduites initiales (point de départ). Ensuite, l'algorithme effectue deux étapes d'affinements successives sur les gaussiennes réduites. Le premier affinement applique l'algorithme k-moyennes pour améliorer les paramètres des gaussiennes réduites. Ensuite, le second affinement se base sur la descente du gradient ainsi que la programmation linéaire pour optimiser davantage les paramètres des gaussiennes réduites.

Le choix du GMRC est justifié par le fait qu'il se distingue parmi les autres algorithmes

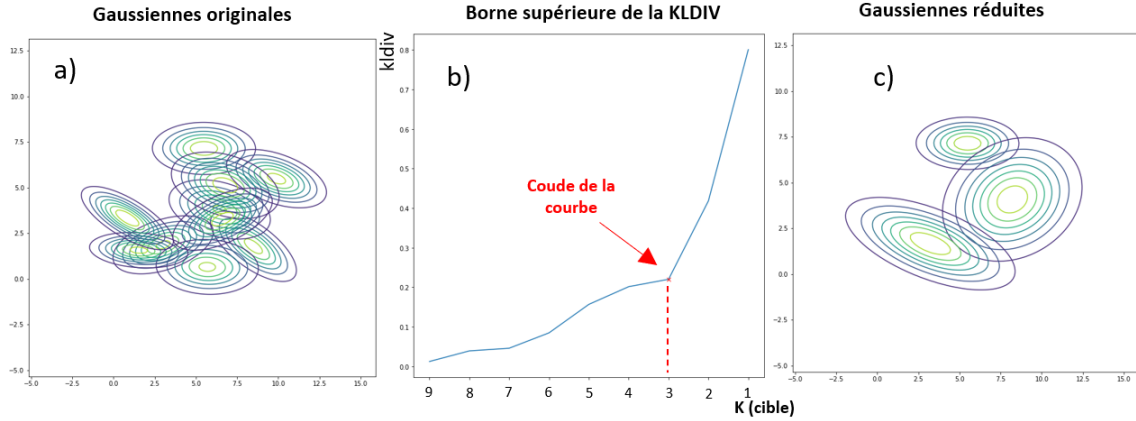


FIGURE 5.5 – Exemple d’estimation du nombre de gaussiennes cible pour une réduction de 10 gaussiennes bi-variées (2D) : a) 10 Gaussiennes originales, b) Évolution de la borne supérieure de la KLD suivant le nombre de gaussiennes cible ( $K$ ) en appliquant l’algorithme de Runnals (algorithme 1), c) gaussiennes réduites en appliquant l’algorithme GMRC avec le nombre de gaussiennes cible estimé ( $K = 3$ )

en matière de performances (qualité de réduction, temps d’exécution) [54]. De plus, l’algorithme se base essentiellement sur la divergence de Kullback (KLD) qui est considérée comme la meilleure mesure pour la réduction de mixture [143, 54]. Enfin, le GMRC adopte un comportement inclusif lors de la réduction [38]. En effet, l’algorithme tente de retenir équitablement un maximum d’informations sur toutes les gaussiennes et ne pas privilégier les gaussiennes majoritaires (en termes de leurs poids). Ce comportement est souhaité dans notre architecture car nous devons retenir les connaissances équitablement sur chacun des contextes et ne pas privilégier les contextes les plus actifs. En effet, les défauts sont rares sur productions industrielles, conduisant à jeux de données très déséquilibrés. Pour autant être, ils doivent tous être correctement détectés.

**Choix du nombre de gaussiennes cible** L’application de tout algorithme de réduction de gaussiennes nécessite de définir a priori le nombre de gaussiennes cible comme entrée de l’algorithme. À notre connaissance, aucun travail permettant d’estimer le nombre de gaussiennes cible n’a été proposé. Nous proposons donc une heuristique permettant de le faire. Cette dernière est basée sur la borne supérieure de la divergence de Kullback (KLD) qui est calculée dans chaque itération de l’algorithme de Runnals (algorithme 1, étape 2, cf. section 2.3.5). Par analogie, cette technique est utilisée dans l’algorithme k-means [100] pour sélectionner le nombre de clusters cible (avec la méthode du coude).



---

**Algorithme 4** : Algorithme de réduction de signatures par GMRC

---

**Input** : Vecteurs moyenne, matrices de covariance et poids de chaque signature

$$Sign_j^i : \mu_j^i, \Sigma_j^i, \pi_j^i, \quad j \in \{1, \dots, k\}, \quad i \in \{1, \dots, n_j\}, \quad n = \sum_{j=1}^k n_j$$

1. Appliquer l'algorithme de Runnals (algorithme 1, section 2.3.5) avec un nombre de composantes cible égal à 1 (i.e.  $K_b = 1$ ), et retenir à chaque itération le coût de la fusion (i.e. borne supérieure de la KLD,  $c_{ij}$  dans l'étape 2 de l'algorithme) le plus faible.
2. Appliquer l'heuristique proposée (coude de la courbe) sur les valeurs du coût de la fusion ( $c_{ij}$ ) retenues dans l'étape 1, pour estimer le nombre de gaussiennes cible ( $K^*$ ).
3. Appliquer l'algorithme de Runnals (algorithme 1, section 2.3.5) avec un nombre de composantes cible égal à  $K^*$  (i.e.  $K_b = K^*$ )
4. Appliquer le premier affinement par k-moyennes (algorithme 2)
5. Appliquer le second affinement (section 2.3.5) par la descente du gradient pour optimiser les vecteurs moyenne et matrices de covariance, ainsi que la programmation linéaire pour optimiser les poids.

**Output** : Vecteurs moyenne, matrices de covariances et poids de chaque signature réduite  $Sign_l^q : \mu_l^q, \Sigma_l^q, \pi_l^q$ . Tel que  $l = 1$  ou  $l = C$ ,  $q \in \{1, \dots, m\}, m < n$ . Si  $l = 1$  alors réduction intra-contexte, sinon généralisation inter-contexte.

---

Nous proposons dans un premier temps d'appliquer l'algorithme de Runnals (algorithme 1) en fixant le nombre de gaussiennes cible à 1. On obtient ainsi une courbe décrivant l'évolution de la borne supérieure de la valeur de la KLD suivant tous les nombres possibles des gaussiennes cibles. Ensuite, nous considérons un accroissement brusque cette valeur comme un indicateur de grande perte d'information et un critère d'arrêt potentiel. Cet accroissement est étroitement lié au coude de la courbe. Par conséquent, nous fixons le nombre de gaussiennes cible à l'abscisse associé au coude de la courbe de l'évolution de la borne supérieure de la KLD.

Une fois le nombre de gaussiennes cible estimé, nous appliquons l'algorithme GMRC avec le nombre de gaussiennes cible estimé. La description algorithmique complète est décrite par l'algorithme 4 et un exemple est présenté sur la figure 5.5.

Dans cette section, nous avons proposé une solution qui permet de répondre aux besoins formulés dans la section 5.1. En effet, deux mécanismes de généralisation ont été proposés. Le premier mécanisme (i.e. réduction intra-contexte) consiste à réduire le nombre de signatures issues d'un même contexte. Le second mécanisme (i.e généralisation

inter-contexte) consiste à définir des signatures généralisées à partir des signatures issues de contextes similaires. Les deux mécanismes proposés consistent à réduire le nombre de signatures similaires tout en retenant le maximum d'information. De ce fait, nous avons proposé un algorithme de réduction de signatures basé sur le GMRC afin d'implémenter les deux mécanismes. La section suivante est dédiée au test et à la validation de la solution proposée.

## 5.3 Expérimentations

### 5.3.1 Protocole expérimental

**Jeux de données** Pour les expériences visant à tester et valider les deux mécanismes proposés, les mêmes jeux de données qui ont été pré-traités dans la section 4.2.1 seront utilisés. De plus, un autre jeu de données issu des mesures de vibration sur des roulements de l'université de Case Western Reserve (CWRU) [151] est inclus. Ce dernier est très largement utilisé dans les travaux sur l'Adaptation au Domaine pour la détection de défauts, en maintenance conditionnelle de machine tournante [164, 97]. Les données de vibration ont été enregistrées pour des charges de moteur de 0 à 3 chevaux-vapeur (vitesses de moteur de 1797 à 1720 tr/min). Par conséquent, 4 jeux de données (1 jeu de données pour chaque charge de moteur) sont obtenus. Le nombre de classes de défauts est égal à 10 sur tous les jeux de données. Pour le pré-traitement de ces derniers, la méthode décrite par Wang et al [164] est utilisée pour constituer les observations sous forme de séries temporelles. En effet, à l'issue de cette étape, 4000 séries temporelles (observations) seront définies et réparties équitablement sur les 10 défauts enregistrés. En d'autres termes, 400 séries temporelles (observations) pour chaque défaut enregistré sont obtenues. Puis, 8 mesures statistiques sont calculées selon [36] pour chaque série temporelle (dans le domaine temporel). Ainsi, pour chaque charge de moteur (i.e, pour chaque jeu de données), un jeu de données de 4000 observations (8 caractéristiques par observation, dimension  $4000 \times 8$ ) est constitué. Les observations (i.e, instance) sont réparties équitablement sur les 10 défauts enregistrés (400 observations par défaut).

**Définition des contextes** Pour la définition des contextes, chaque charge de moteur est considéré comme un *Contexte* dans le jeu le donnée CWRU. Par conséquent, 4000 ob-

servations (400 par défaut) sont enregistrées dans chacun des 4 contextes. Pour les autres jeux de données, seules les données provenant du jeu "Hydro" peuvent servir à définir des Contextes. Les autres jeux de données n'ont aucun attribut ou manière permettant de définir des contextes.

Pour les jeux de données "hydro", notons que chaque opération (observation) a une durée de 6s et que durant l'opération, 17 capteurs enregistrent les mesures sous forme de séries temporelles avec 3 fréquences d'échantillonnage différentes (10, 100, 1000 Hz) selon le capteur. En visualisant la série temporelle associée au capteur PS1 (figure 5.6), on observe qu'il existe 6 paliers successifs distincts, toutes les secondes, indiquant probablement un changement de paramètres opératoires, et donc de Contexte. Par conséquent, nous avons divisé la série temporelle en 6 parties et nous considérons que chaque partie est un Contexte. Par exemple, les mesures associées à la première seconde (0-1 seconde) sur tous les capteurs sont prises pour former le jeu de donnée associé au premier contexte. Ainsi, 6 Contextes sont définis pour les jeux de données "hydro". Le pré-traitement des séries temporelles sur chacun des contextes reste identique à celui de la section 4.2.1 ainsi que tous les autres paramètres des expériences de la section 4.2.3.

Les jeux de données qui découlent de cette partie seront donc utilisés que pour la validation de la généralisation inter-contexte (section 5.3.3).

### **5.3.2 Mécanisme 1 : Réduction intra-contexte**

Dans cette partie, l'apprentissage itératif qui a été décrit dans la partie 4.2.3 est repris pour la validation de la robustesse de l'architecture. Rappelons que l'idée générale de l'expérience consiste à effectuer un peuplement de l'ontologie sur une petite partie du jeu de données d'apprentissage, où la distribution des données de défauts est aléatoire. Ensuite, l'enrichissement itératif de l'ontologie est enclenché sur l'autre partie du jeu de données d'apprentissage, avec à chaque étape d'enrichissement, des métriques de performances sont mesurées sur le jeu de données de test. Les jeux de données présentés dans la section 4.2.1 sont utilisés pour les expériences. En effet, comme la réduction se fait uniquement sur un seul contexte, il n'y a pas besoin de définir plusieurs contextes. L'objectif est de montrer que la réduction améliore (ou a minima ne détériore pas) le système de détection de défauts. À cet effet, nous proposons deux expériences de réduction : une réduction post-apprentissage, une réduction pendant l'apprentissage (à chaque itération).

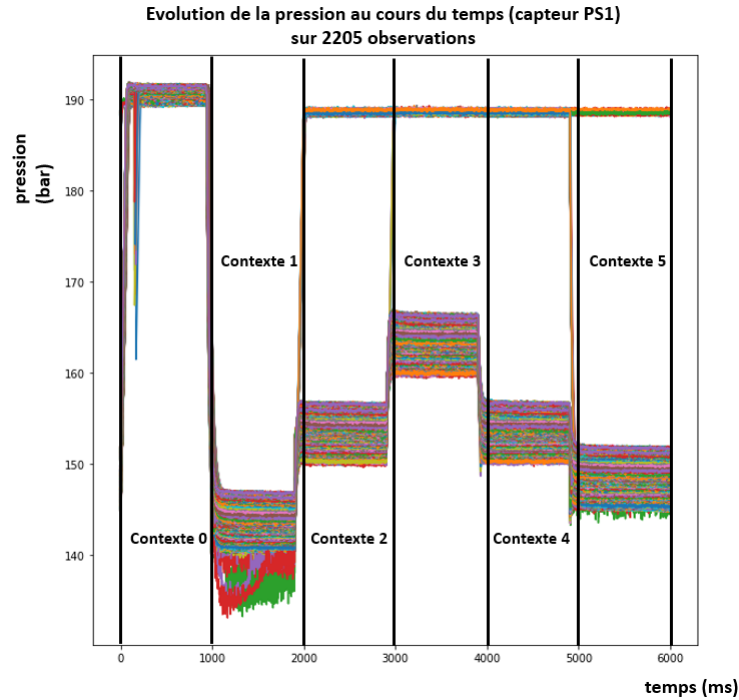


FIGURE 5.6 – Description de la série temporelle associée au capteur PS1 pour les 2205 observations du jeu de donnée "hydro", et définition des contextes (lignes verticales)

Il convient de rappeler que nous effectuons une réduction pour chaque défaut séparément comme décrit dans la section 5.2.2.

**Réduction post-apprentissage** Dans cette partie, nous reprenons précisément les gaussiennes qui ont été apprises dans chaque étape d'enrichissement de la partie (résultats des figures 4.9). Puis, pour chaque étape d'enrichissement, nous effectuons une réduction des gaussiennes avec l'algorithme GMRC (en appliquant la méthode du coude à la borne supérieur de la KLD pour estimer le nombre de composantes cible du GMM, cf. figure 5.5). Enfin, nous classifions les observations du jeu de données test avec les gaussiennes réduites. Mais pour l'itération suivante, c'est bien le résultat de la figure 4.9 qui est alors réduit ; l'itération ne repart pas du modèle réduit précédent (comme ce sera le cas dans la prochaine série de tests). Les résultats comparant les deux approches (avec ou sans réduction) sont illustrés sur la figure 5.7.

Globalement, nous observons une amélioration des métriques lorsque la réduction est effectuée. En effet, le taux de reconnaissabilité est meilleur sur toutes les étapes d'enrichissement comparé à l'approche originale (sans réduction, figure 4.9). Cela est montré par le

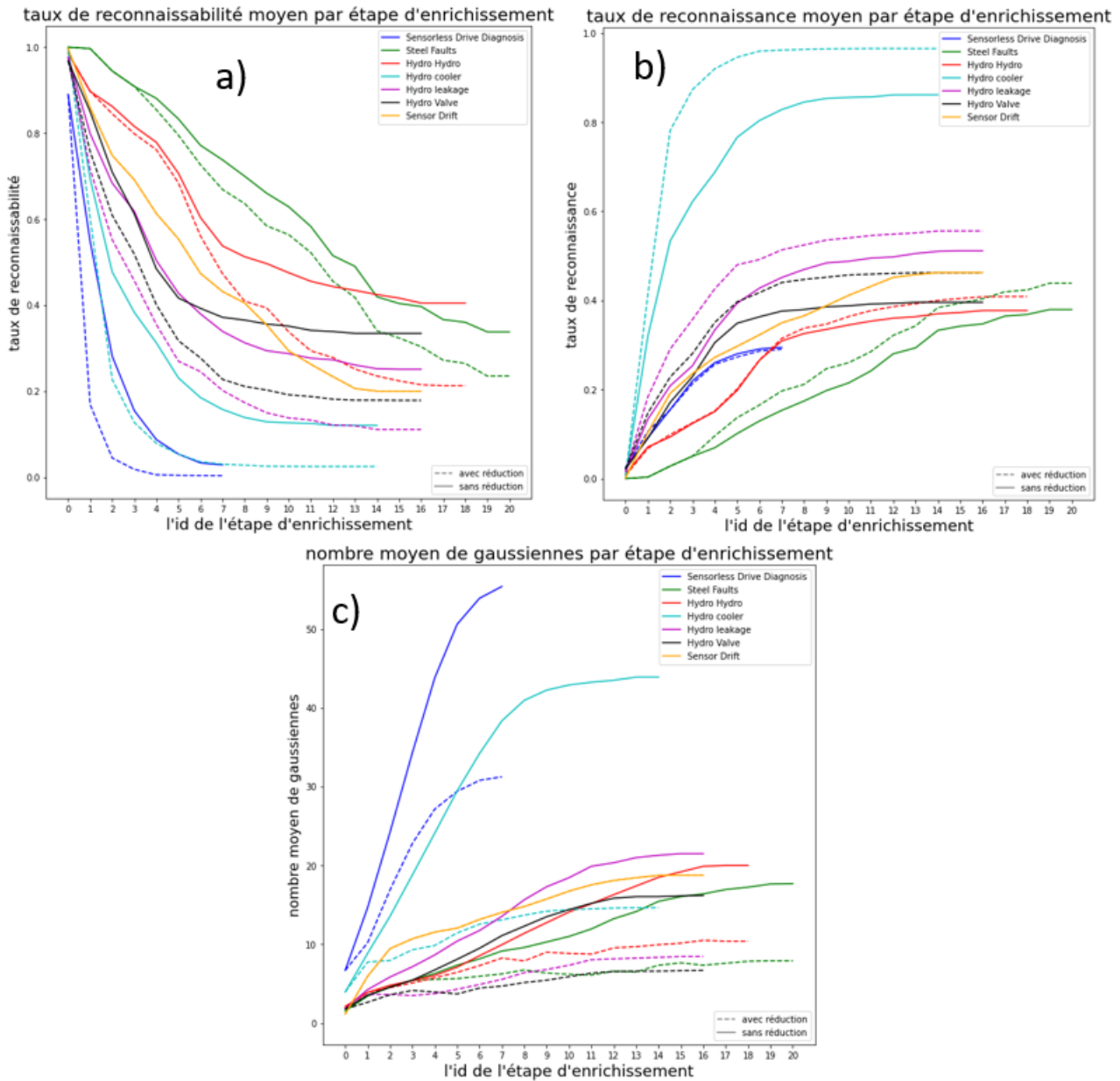


FIGURE 5.7 – Réduction post-apprentissage : a) taux de reconnaissance, b) taux de reconnaissance, c) nombre moyen de gaussiennes, dans le peuplement initial de l'ODD (step 0) et pendant le processus itératif de découverte de connaissances, en moyennant sur 20 expériences différentes

fait que les courbes associées à la réduction sont toutes en dessous de leurs courbes respectives dans le scénario sans réduction. De même, le taux de reconnaissance est nettement supérieur avec les gaussiennes réduites (inversement, les courbes avec réduction sont au dessus de leurs courbes respectives). Enfin, le nombre de gaussiennes est nettement moins

important, ce qui est rassurant comme c'était l'objectif principal de l'algorithme proposé (figure 5.7.c). En résumé, nous obtenons de meilleurs résultats de classification avec un nombre de gaussiennes moins important. Cela peut s'expliquer par le fait que la réduction de gaussiennes diminue le sur-apprentissage cumulé, en fusionnant des gaussiennes similaires. Ainsi, notre proposition améliore globalement la capacité généralisation de notre architecture par le biais de la réduction de gaussiennes, avec de surcroît de meilleures performances de classification.

**Réduction pendant l'apprentissage** Dans cette partie, nous reprenons également la même expérience décrite dans la section 4.2.3. En revanche, la différence est que nous appliquons une réduction de gaussiennes après chaque étape d'enrichissement durant l'apprentissage itératif; et qu'ensuite, nous continuons l'apprentissage itératif avec les gaussiennes réduites de l'itération précédente. Ce processus est répété pour chaque étape d'enrichissement jusqu'à l'épuisement du jeu de donnée d'apprentissage. Les résultats sont illustrés sur la figure 5.8.

Comme pour l'expérience précédente, nous observons que les taux de reconnaissances et de reconnaissabilités sont clairement supérieures lorsque la réduction de modèle est appliquée en cours d'apprentissage. De même pour le nombre de gaussiennes qui est nettement moins élevé (figure 5.8.c). Enfin, nous observons également que le nombre d'étapes d'enrichissement est moins important. Cette capacité à apprendre plus rapidement est très intéressant d'un point de vue opérationnel. En effet, la production sera mieux suivie plus tôt, et moins d'observations de défauts sera nécessaire pour cela. Cela s'explique par le fait que l'architecture reconnaît davantage d'observations durant de l'apprentissage itératif (figure 5.8.a) lorsque la réduction est appliquée (par le biais de la réduction du sur-apprentissage). De ce fait, l'ensemble des rejets est beaucoup moins alimenté durant l'apprentissage itératif, ce qui explique le nombre réduit d'étapes d'enrichissements. De plus, les gaussiennes apprises sur l'ensemble des rejets sont de meilleure qualité (figure 5.8.b). Globalement, la réduction améliore les performances de notre système de détection de défauts tout en minimisant le nombre d'étapes d'enrichissements et le nombre de gaussiennes.

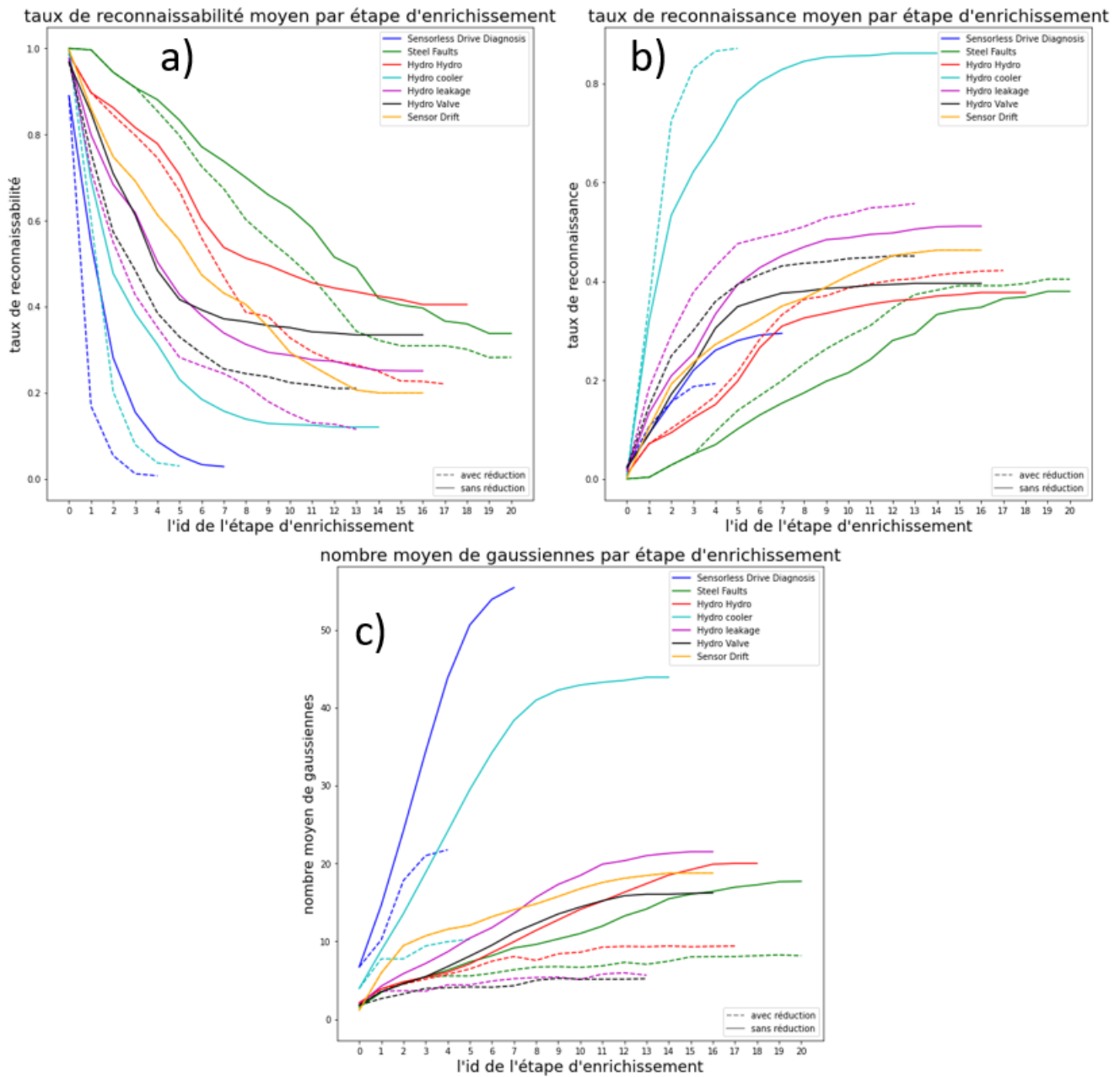


FIGURE 5.8 – Réduction pendant l'apprentissage : a) taux de reconnaissabilité, b) taux de reconnaissance, c) nombre moyen de gaussiennes, dans le peuplement initiale de l'ODD (step 0) et pendant le processus itératif de découverte de connaissances, en moyennant sur 20 expériences différentes

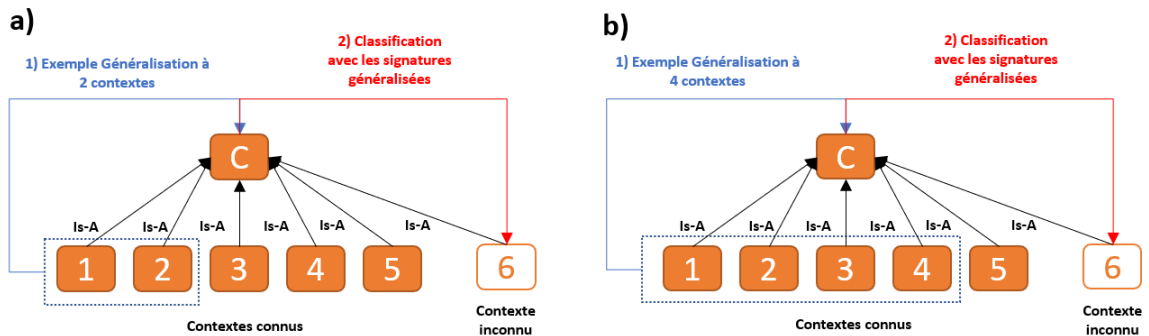


FIGURE 5.9 – Exemples de généralisation sur 5 contextes connus : a) exemple de généralisation sur 2 contextes, b) exemple de généralisation sur 4 contextes

### 5.3.3 Mécanisme 2 : Généralisation inter-contextes

Dans cette partie, les jeux de données associés aux contextes identifiés et définis au 5.3.1 (Hydro et CWRU) seront utilisés. L’objectif est de classifier les observations provenant de contextes inconnus en réutilisant les connaissances déjà capturées dans l’ontologie. Pour ce faire, deux expériences sont envisagées : la généralisation ponctuelle et la généralisation dans l’apprentissage itératif.

**Généralisation ponctuelle** Dans cette expérience, l’intérêt de la généralisation est investigué sur un apprentissage non itératif. L’objectif est de montrer que la généralisation des signatures permet la reconnaissance des défauts dans des contextes inconnus. Le choix de l’apprentissage ponctuel est justifié par le fait que l’objectif, dans un premier temps, est de montrer que la généralisation est plus performante qu’une simple utilisation des signatures des autres contextes connus. De plus, le second objectif consiste à étudier la qualité de la généralisation en fonction du nombre de contextes pris. À cet effet, il est plus simple d’approcher cet objectif en adoptant un approche ponctuelle sans besoin d’inclure un apprentissage itératif qui induit une complexité supplémentaire inutile. De plus, comme l’objectif de la généralisation est de classifier les observations sur des inconnus à partir des signatures généralisées, les mesures de performance sur les contextes connus ne sont pas retenues (i.e. qui ont servi à la généralisation de signatures).

À cet effet, nous prenons le jeu de données associé à chacun des contextes et le divisons en jeu d’apprentissage et de test (80%/20%). Ensuite, nous apprenons des signatures sur les observations associées à chacune des classes présentes dans le jeu de données d’entraînement, i.e. un GMM en supervisé. Nous aurons donc des signatures associées à



chaque classe de défauts sur tous les contextes.

Afin de tester la généralisation ponctuelle, nous allons effectuer la généralisation sur toutes les combinaisons de contextes connus. Un exemple est illustré sur la figure 5.9. En effet, en considérant le 6<sup>ème</sup> contexte inconnu et les 5 premiers connus, nous pouvons effectuer une généralisation à 2 contextes (figure 5.9.a) avec un nombre de  $\mathcal{C}_2^6 = 15$  combinaisons possibles. De même nous pouvons aussi effectuer une généralisation à 4 contextes (figure 5.9.b) avec un nombre de  $\mathcal{C}_4^6 = 15$  combinaisons possibles. Les signatures généralisées serviront à classifier les observations dans les contextes inconnus.

Pour chaque combinaison sur un nombre de contextes fixé, nous mesurons les métriques (taux de reconnaissance et de reconnaissabilité) uniquement sur les contextes n'ayant pas servi à la généralisation. Par exemple, en généralisant les signatures issues des contextes (1, 2), nous allons les utiliser pour classifier les observations des jeux de données de test associées aux contextes (3, . . . , 6) en mesurant les métriques pré-déterminées. Ensuite, nous réalisons la même expérience sur toutes les combinaisons possibles à deux contextes. À l'issue de toutes les généralisations à 2 contextes, nous aurons plusieurs taux de reconnaissabilité et de reconnaissance pour chaque contexte. Nous pouvons donc calculer la moyenne de ces métriques pour chacun des contextes. Nous aurons ainsi un taux de reconnaissabilité et de reconnaissance moyen pour chaque contexte sur une généralisation à deux contextes. Enfin, nous répétons ces expériences sur toutes les combinaisons de 1 à 5 contextes (dans le cas où nous avons 6 contextes, car nous devons avoir au minimum 1 contexte inconnu).

Par ailleurs, afin de comparer les résultats, il convient de souligner que nous répétons les mêmes expériences décrites ci-dessus à la différence que la généralisation s'effectue avec une simple agrégation de signatures. Cette agrégation revient simplement à instancier des copies des signatures spécifiques aux contextes pris pour la généralisation, et de les relier au concept général (C) englobant les 6 contextes (figure 5.9). Nous illustrons seulement les figures associées aux jeux de données "Hydro Cooler" (figure 5.10) et "CWRU" (figure 5.11), les autres figures sont fournies en annexe 6.5.

Nous observons sur la figure 5.10 ainsi que les figures en annexe 6.5 que la généralisation avec réduction est nettement plus performante qu'une à une simple agrégation de signatures entre différents contextes similaires. En effet, nous observons que les taux de reconnaissabilité de la généralisation avec réduction sont très faibles par rapport à ceux utilisant une simple agrégation. Inversement, les taux de reconnaissances sont bien

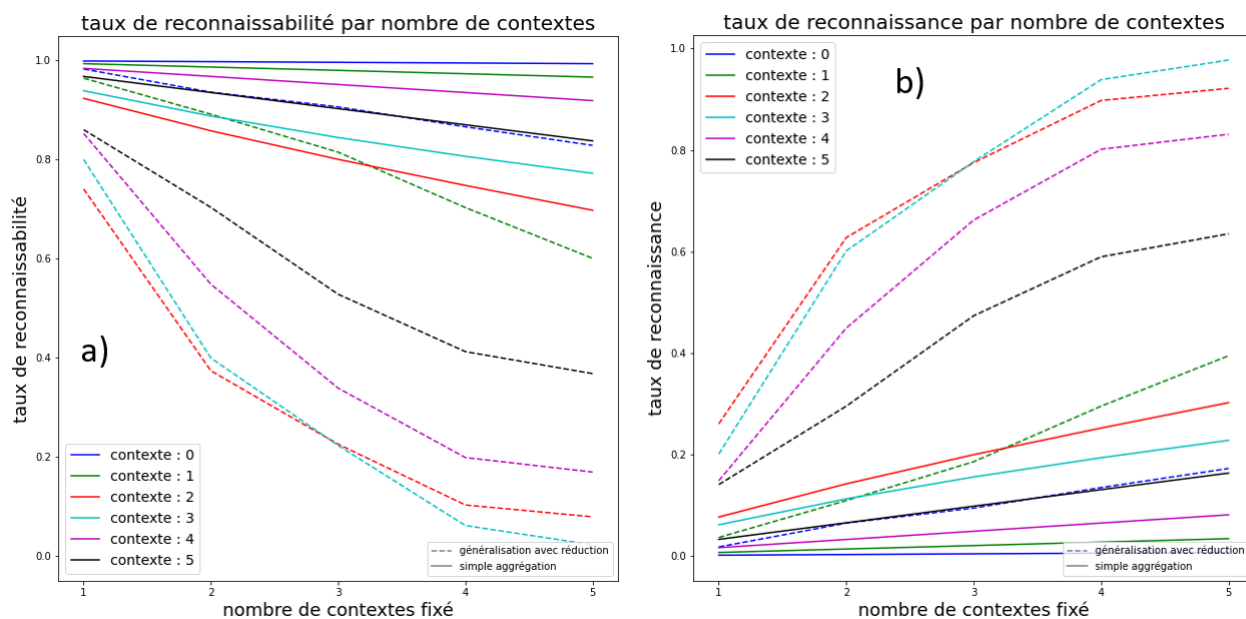


FIGURE 5.10 – Résultats de généralisation ponctuelle par nombre de contexte sur Hydro Cooler : a) taux de reconnaissabilité b) taux de reconnaissance

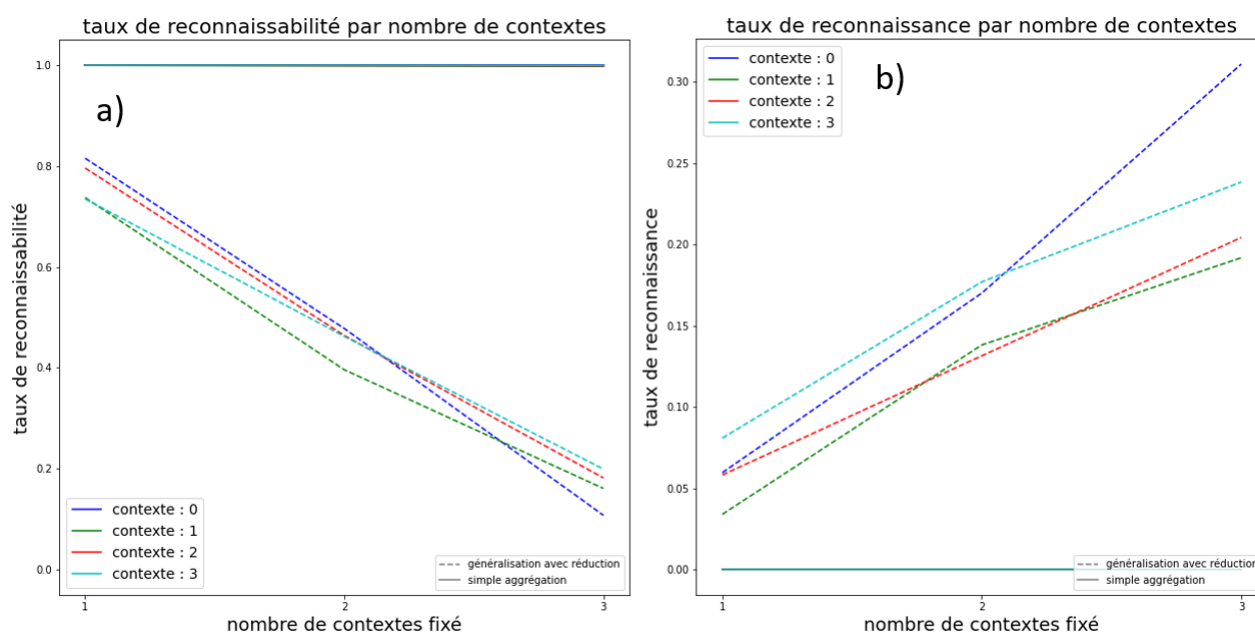


FIGURE 5.11 – Résultats de généralisation ponctuelle par nombre de contexte sur CWRU : a) taux de reconnaissabilité b) taux de reconnaissance

meilleurs lorsque la généralisation avec réduction est appliquée. Enfin, nous visualisons que ces métriques s'améliorent au fur et à mesure qu'on inclut plus de contextes lors

de la généralisation. En d'autres termes, plus nous avons de contextes connus à notre disposition, mieux seront les performances de notre système de détection défauts dans les contextes inconnus similaires. Par conséquent, la généralisation peut améliorer notre système itératif de détection de défauts.

Le seul contre exemple est illustré sur la figure 5.11. En effet, même si les métriques s'améliorent en fonction du nombre de contextes, nous observons que le taux de reconnaissance ne dépasse pas les 30% alors que les taux de reconnaissabilité sont très faibles (presque 10%). Nous donnons plus d'explications dans la partie discussion 5.3.4.

**Généralisation appliquée à l'apprentissage itératif** Dans cette partie, le schéma itératif décrit dans la section 4.2.3 est repris. Nous supposons dans un premier temps que des signatures issues de plusieurs contextes ont été apprises et capturées par l'ontologie. Ensuite, nous simulons dans un second temps, un arrivage en temps réel d'observations issues de contextes inconnus. L'objectif est donc de traiter ces observations. À cet effet, il existe 3 manières de les traiter.

Le premier scénario est de rejeter toutes les observations associées aux contextes inconnus. Puis, apprendre des premières signatures sur le jeu de données d'exception une fois que leurs nombres aura atteint le seuil d'apprentissage fixé sur l'ensemble de rejets. Ceci revient à ne pas tenir compte de connaissances apprises dans d'autres contextes. Le second et troisième cas est de démarrer respectivement la classification en temps réel avec les signatures issues d'une généralisation par une simple agrégation ; ou bien à partir des signatures généralisées issues d'une généralisation avec réduction (tels que tous deux définis au cours de l'expérience précédente). Dans ces deux cas, ces signatures généralisées serviront seulement à initier l'apprentissage incrémental dans le contexte inconnu. Ensuite, l'apprentissage incrémental est réalisé suivant le schéma classique de la section 4.1.3 (i.e, rejet et enrichissement) avec réduction pendant l'apprentissage.

Durant les expériences, nous supposons qu'un seul contexte est inconnu et que tous les autres sont connus. De ce fait, il convient dans un premier temps d'apprendre les signatures associés aux défauts des contextes supposés connus. À cet effet, nous appliquons le procédé itératif (section 4.2.3) avec réduction pendant l'apprentissage afin des les apprendre. Une fois toutes les signatures apprises sur les contextes connus, nous simulons l'arrivage en temps réel des observations associées au contexte inconnu.

Ainsi, sur un jeu de données à 6 contextes, nous aurons donc 6 expériences à effectuer

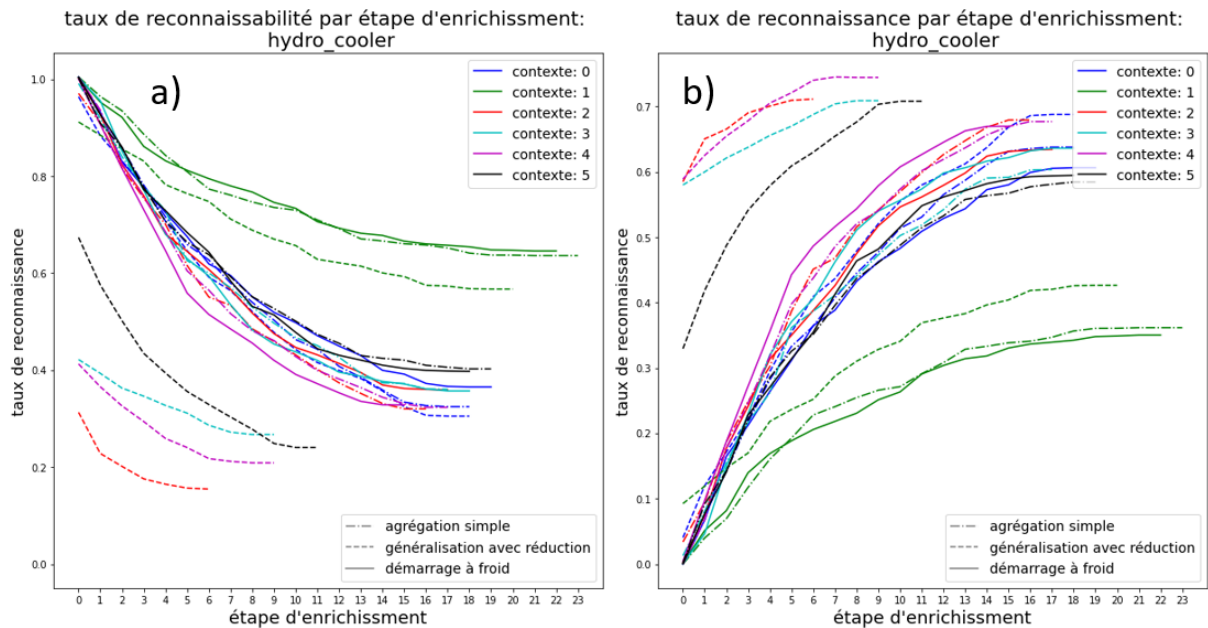


FIGURE 5.12 – Résultats de la généralisation appliquée à l'apprentissage itératif sur Hydro Cooler : a) taux de reconnaissabilité b) taux de reconnaissance

(1 expérience par contexte). Par exemple, en considérant le 6<sup>ème</sup> contexte inconnu et les 5 premiers connus, nous commençons par appliquer le processus (section 4.2.3) séparément pour chacun des contextes afin d'apprendre de façon itérative les signatures dans chacun des 5 premiers contextes (avec réduction). Ensuite, une fois toutes les signatures apprises sur les contextes connus, nous simulons l'arrivage des données du 6<sup>ème</sup> contexte. Nous répétons cette expérience en considérant à chaque fois qu'un seul contexte est inconnu et que tous les autres sont connus (nous aurons donc 6 expériences pour un lot de 6 contextes).

À l'arrivée des observations sur le contexte inconnu, nous appliquons séparément les 3 méthodes énumérées ci-dessus pour traiter ces observations. La comparaison des résultats issus des 3 méthodes est illustrée dans les résultats. De même que précédemment, nous visualisons seulement les figures associées aux jeux de données "Hydro Cooler" (figure 5.12) et "CWRU" (figure 5.3.3). Les autres sont fournies en annexe 6.5.

Nous observons à travers la figure 5.12 que la généralisation avec réduction donne de meilleurs résultats en termes de taux de reconnaissance et de reconnaissabilité. Cette amélioration est expliquée par le fait que l'essentiel des connaissances apprises sur chacun des contextes a été remonté dans le concept général tout en réduisant le sur-apprentissage.

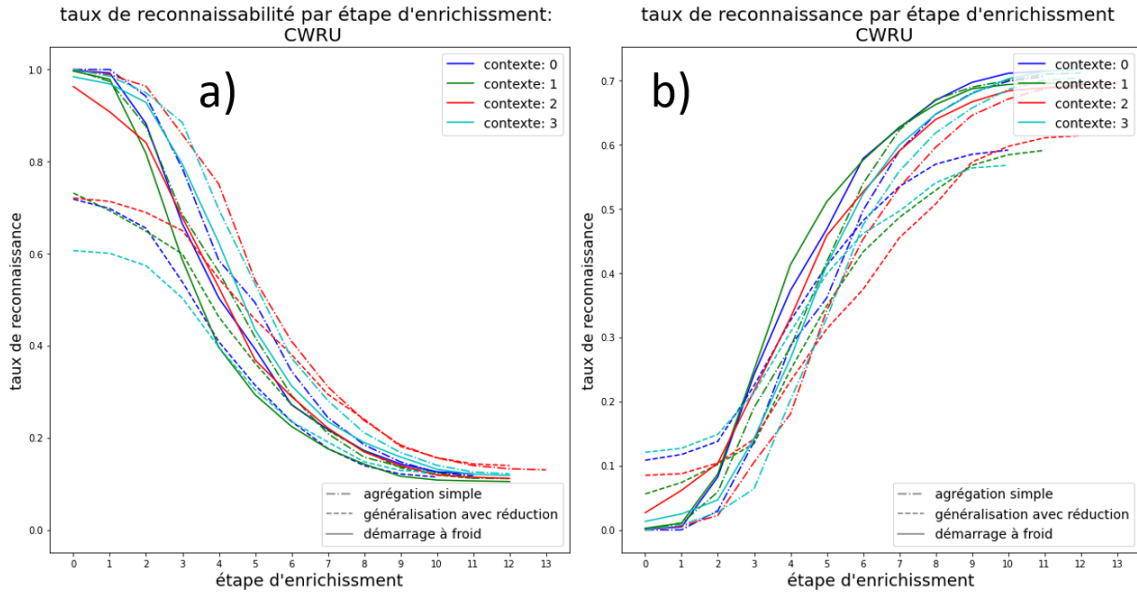


FIGURE 5.13 – Résultats de la généralisation appliquée à l'apprentissage itératif sur CWRU : a) taux de reconnaissabilité b) taux de reconnaissance

De même, nous remarquons que le nombre d'étapes d'enrichissement est nettement moins important lorsque la généralisation avec réduction est effectuée. Ainsi, nous obtenons de meilleurs résultats avec moins d'apprentissages. Enfin, nous remarquons que l'agrégation de signatures n'apporte quasiment pas d'information et est équivalente à un démarrage à froid (sans connaissance de signatures dans d'autres contextes), d'où l'intérêt de la généralisation avec réduction. Les mêmes constats sont valides pour les autres jeux de données en annexe 6.5.

Le seule limitation est illustrée sur le jeu de donnée "CWRU". En effet, nous remarquons au début de l'apprentissage que les métriques associées à la généralisation avec réduction sont meilleures que celles des 2 autres méthodes. En revanche, ces dernières deviennent relativement moins bonnes au fur à mesure de l'apprentissage en terme de taux de reconnaissance. Les taux des reconnaissabilités quant à eux convergent vers le même point final. Nous donnons plus d'explications à ce phénomène dans la partie suivante 5.3.4.

### 5.3.4 Discussions et pistes d'amélioration

Nous avons montré que la généralisation permet d'améliorer considérablement le système de détection de défauts. En effet, la généralisation intra-contexte permet d'augmenter

la capacité de détection de défauts tout en réduisant le nombre de signatures. De plus, le fait d'exploiter les connaissances apprises dans des contextes similaires permet de reconnaître les mêmes défauts qui ont été appris précédemment (inter-contexte). En revanche, nous avons constaté par ailleurs que la généralisation peut parfois détériorer le système de détection de défauts quand elle est appliquée sur certains jeux de données (CWRU). Afin d'étudier cela, nous nous basons sur les figures 5.11 et 5.3.3, relatives à CWRU.

Nous constatons à travers ces figures que le taux de reconnaissabilité diminue au fur à mesure que nous introduisons des contextes (figure 5.11) et de même pour l'apprentissage itératif en fonction des étapes d'enrichissement (figure 5.3.3). Nous concluons donc que les signatures généralisées permettent de reconnaître les observations issues des contextes inconnus. En revanche, nous remarquons que les taux de reconnaissance restent relativement faibles en dépit de leur croissance. Ce constat nous indique que les signatures généralisées affectent les observations dans de mauvaises classes, sur ce jeu de données CWRU. Par conséquent, nous pouvons penser que les signatures généralisées de différents défauts sont en fait superposées. Ce qui explique qu'elles arrivent à reconnaître les observations (ne pas les rejeter) mais ne les affectent pas dans les bonnes classes. Ce phénomène mène à un faible taux de rejet lors de la phase d'apprentissage itératif (figure 5.3.3) et nous apprenons moins de signatures spécifiques à des classes, par rapport à un démarrage à froid, où l'on rejette beaucoup d'observations durant l'apprentissage itératif.

Pour valider notre hypothèse, nous avons appliqué une Analyse en Composantes Principales (ACP) pour visualiser la distribution de deux défauts sur deux contextes sur les jeux de données Hydro Cooler et CWRU (figure 5.14). L'ACP a été réalisée sur un jeu de donnée concaténant les observations des deux défauts sur les deux contextes pour chaque source de données (Hydro Cooler et CWRU). L'objectif est de vérifier visuellement la superposition des défauts dans l'espace des caractéristiques. Nous pouvons constater à travers la figure 5.14 qu'il y a une superposition des distributions de défauts assez importante dans CWRU contrairement à Hydro Cooler où les classes de défauts sont séparées. De ce fait, il existe un risque important que les signatures généralisées d'un défaut soient superposées à d'autres distributions de défauts dans CWRU. Ainsi, la généralisation risque de détériorer le système de détection de défauts dans le cas où la distribution des données est similaire à celle de CWRU.

Afin d'y remédier, une solution pourrait consister à modifier les caractéristiques qui ont été prises pour le pré-traitement (section 5.3.1) afin d'avoir une meilleure séparation

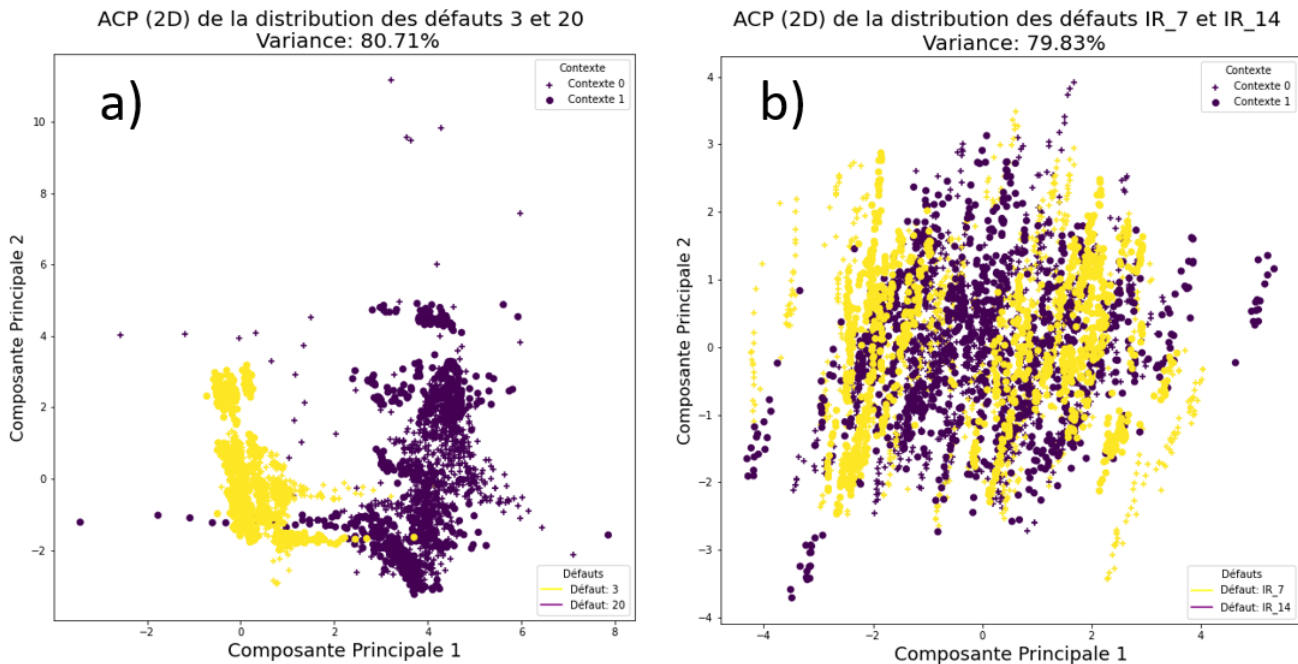


FIGURE 5.14 – ACP (2D) de la distribution des défauts sur 2 contextes : a) Hydro Cooler  
b) CWRU

entre les classes de défauts (comme dans Hydro Cooler). En revanche, si cette solution ne peut pas être envisagée, il conviendrait d’alerter l’utilisateur que la généralisation risque de détériorer le système.

Pour ce faire, nous pouvons comparer les signatures de défauts issue de la généralisation. En effet, l’objectif est de détecter une superposition importante entre les signatures généralisées de défauts différents. À cet effet, la divergence de Kullback entre les signatures inter et intra-défauts peut être mesurée. La divergence intra-défaut indique à quel point les signatures généralisées d’un même défaut sont proches. La divergence inter-défaut indique à quel point les signatures généralisées de différents défauts sont éloignées. Ainsi, si les mesures inter-défauts sont inférieures à celles de l’intra-défauts, nous aurons une forte indication que les signatures généralisées de plusieurs défauts sont superposées. Cette indication permettrait d’alerter l’expert du risque de détérioration du système de détection défaut, et par conséquent, lui recommander trois actions. La première consiste à lui proposer une fusion des deux types de défauts détectés s’il s’avèrent être de la même famille. La deuxième lui propose de rejeter cette généralisation, s’il tient à pouvoir distinguer ces 2 types de défauts. La dernière consiste à lui proposer de retravailler sur ses caractéristiques, pour mieux distinguer ces 2 défauts. Nous laissons ce travail en

perspective.

Par ailleurs, un autre cas peut se présenter dans le sens où des distributions de données issues de contextes similaires (d'un point de vue ontologie), sont réalité très éloignées. Une généralisation de signatures effectuée sur l'ensemble de ces données peut définir des signatures généralisées qui soient de mauvaise qualité, ce qui entraîne une dégradation de la performance de l'architecture. Théoriquement, l'algorithme de GMRC soutenu par l'heuristique du coude permettra de détecter l'écart entre les distributions et arrêtera la généralisation inter-contexte. En revanche, il convient de tester cette hypothèse en pratique pour étudier la sensibilité de l'architecture quant à l'éloignement des distributions de données inter-défauts. Nous laissons donc ce travail en perspective.

## 5.4 Conclusion

Dans ce chapitre, nous avons proposé la dernière étape de notre architecture de détection de défauts. Nous avons commencé par exprimer les besoins et fait évoluer la structure de l'ontologie. Puis, nous avons proposé deux mécanismes de généralisation de connaissances permettant d'améliorer le système de détection de défauts.

Le premier mécanisme de généralisation consiste à réduire le nombre de signatures sur un contexte déterminé. Cette réduction a pour but de réduire les redondances et obsolescences des signatures. De plus, nous améliorons la capacité de généralisation du système de détection en réduisant le sur-apprentissage. Ce mécanisme a été testé sur deux expériences (post et pendant l'apprentissage) et a fourni des résultats très prometteurs sur 8 jeux de données d'UCI. En effet, le modèle s'apprend plus vite (réduction du nombre d'étapes d'enrichissement de 20% en moyenne) et mieux (augmentation, réduction moyenne de 20% sur la précision et le taux de rejet, respectivement), avec moins de gaussiennes (réduction du nombre de gaussienne de 50% en moyenne).

Le deuxième mécanisme permet de généraliser les signatures spécifiques à plusieurs contextes. Cette généralisation permet de remonter les connaissances sous forme de signatures généralisées vers le concept général englobant un groupe de sous-contextes similaires. Puis, ces dernières serviront à classifier les observations sur des contextes inconnus similaires héritant du concept général. Nous avons montré à travers les expériences sur 6 jeux de données d'UCI, que la généralisation permet d'augmenter considérablement les performances du système de détection. En effet, cela nous a permis de réduire (d'augmenter) de



20% en moyenne le taux de rejet (la précision) respectivement dans la phase de test. De plus, les résultats améliorés sont obtenus avec moins d'étapes d'enrichissement, ce qui est très intéressant d'un point de vue opérationnel. En effet, la production sera mieux suivie plus tôt, et moins d'observations de défauts sera nécessaire pour cela. En outre, les résultats continuent de s'améliorer en fonction du nombre de contextes à généraliser. Ainsi, plus nous avons de contextes à notre disposition sur lesquels nous pouvons généraliser, mieux seront les résultats.

Par ailleurs, nous avons constaté une limite de la généralisation dans un cas où il y a une forte superposition de signatures de défauts dans l'espace de caractéristiques. De ce fait, nous avons proposé une piste d'investigation permettant d'alerter l'utilisateur du risque de la généralisation. Ainsi, ce dernier pourra prendre des actions correctives (modification des caractéristiques par exemple) afin d'améliorer le système de détection de défauts.

Arrivés à la fin de ce chapitre, il convient de prendre un recul sur l'architecture ML-KM proposée. En effet, L'architecture a permis de répondre aux manques identifiés dans l'état de l'art. Nous avons montré à travers les expériences de ce chapitre que l'exploration et la généralisation des connaissances de l'ontologie (KM), permettent d'améliorer la détection de défauts (ML) dans les contextes connus (réduction intra-contextes) ainsi que les contextes inconnus (généralisation inter-contextes). En effet, les expériences ont montré que la détection de défauts est plus précise, et s'améliore davantage lorsque la généralisation des connaissances est appliquée. Nous pouvons donc affirmer à travers les nombreuses expériences (6 à 8 données d'UCI, et à travers les nombreux protocoles expérimentaux) que l'utilisation des connaissances peut améliorer la détection de défauts (ML) dans les procédés de fabrication. De ce fait, il est crucial de capturer plusieurs connaissances pour continuer à améliorer la surveillance.

De l'autre côté, l'acquisition de connaissance est assurée par les modèles de ML utilisés ainsi que la connaissance de l'expert du domaine. En effet, comme il est compliqué en réalité de se baser uniquement sur les connaissances de l'expert, du fait d'un manque d'expérience ou de la qualité d'implication, nous avons essayé de combler ce vide par des modèles de ML. De plus, comme nous avons fourni à ces modèles la capacité d'être sensibles à de nouvelles connaissances qui peuvent apparaître durant la production (i.e. nouveaux défauts), cela donne à notre architecture une capacité évolutive (contrairement aux approches ML traditionnelles) dans le sens où elle est toujours mise à jour avec la

dynamique de la production.

Enfin, il convient de rappeler des méthodes classiques de ML peuvent rapidement atteindre leurs limites du fait de la multitude de contextes opératoires. En effet, il est difficile (parfois impossible) de surveiller l'entièreté du site de production avec un seul modèle. De ce fait, la gestion de connaissances (KM) par une ontologie a permis de réduire la complexité induite par la multitude de contextes opératoires, ce qui permet de faciliter la tâche aux modèles de ML. En effet, une gestion élégante des connaissances découvertes par le ML et de l'expert, a permis de rendre l'architecture facilement implémentable en industrie.

Nous consacrons le dernier chapitre nos contributions à l'application de notre architecture dans le domaine du vissage.



# DECOUVERTE ET GÉNÉRALISATION DE CONNAISSANCES EN VISSAGE

---

Dans ce chapitre, la dernière contribution de notre thèse est présentée. Cette dernière repose sur l'application de nos contributions présentées dans les chapitres 4 et 5 dans le domaine du vissage. Les études de cas ont été développées au sein de la société Desoutter. Les données proviennent d'une base de données collectée lors de l'assemblage de composants de moteurs dans l'industrie automobile.

Ce chapitre est organisé comme suit : la présentation du métier est fournie dans la section 6.1. Ensuite la découverte incrémentale de nouveaux défauts dans la section 6.2 est décrite. La partie généralisation des connaissances est appliquée et illustrée dans la section 6.3. Enfin, une brève présentation de l'implémentation industrielle de la solution est fournie dans la section 6.4.

## 6.1 Présentation du métier

### 6.1.1 Notions de base sur les assemblages boulonnés

Le vissage est une opération qui consiste à assembler deux pièces à l'aide d'un ensemble de vis. Pour ce faire, l'opération commence par créer un logement pour l'insertion de la vis (filet). Puis, il suffit de tourner la vis dans le taraudage puis d'appliquer un couple mécanique. Le couple appliqué met en tension la vis insérée (qui s'allonge comme un ressort) dans son logement et crée une force axiale (tension). Cette tension maintient les pièces assemblées, par adhérence (figure 6.1). Cette dernière doit être assez forte afin de maintenir les deux pièces (et éviter un glissement relatif), et assez faible pour ne pas endommager la vis ou l'assemblage.

Une compréhension approfondie du processus de vissage est essentielle pour une détection fiable et précise des défauts [79]. Typiquement, le processus peut être segmenté en plusieurs étapes à travers la courbe couple-angle (figure 6.2). La première étape correspond à *l'accouplement initial* du filet de la vis dans celui de l'écrou. La deuxième zone est la zone du *pré-vissage*, dans laquelle la vis est entraînée régulièrement dans le trou jusqu'à ce que la tête de vis entre en contact avec la surface d'appui. La troisième zone est la zone *d'alignement*, pendant laquelle les surfaces de contact de la fixation et de la vis s'alignent. La zone d'ajustement non linéaire est une combinaison complexe de macroeffets dus à l'alignement des pièces de l'assemblage et de microeffets dus aux déformations de la surface et des filetages.

La quatrième zone est la zone de *montée en couple*, dans laquelle la pente de la courbe

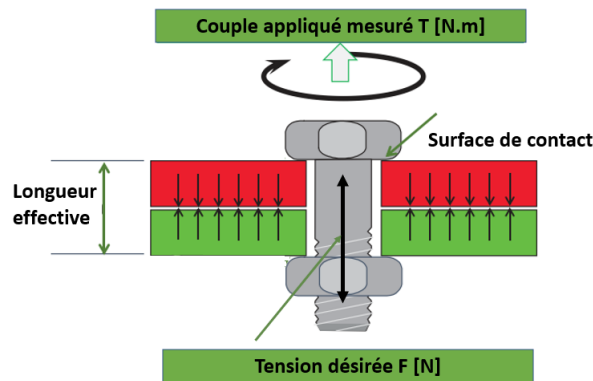


FIGURE 6.1 – Description du vissage [79]

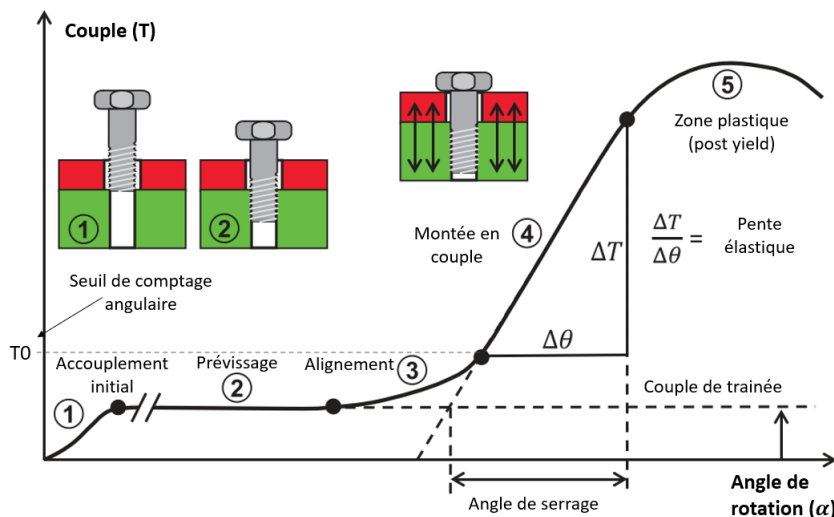


FIGURE 6.2 – Courbe couple-angle durant l'opération de vissage [79]

couple-angle est constante. C'est dans cette phase que la tension qui maintient les pièces s'accroît progressivement. De ce fait, il est nécessaire d'appliquer le bon couple, pour obtenir la tension souhaitée. Il existe une relation linéaire entre la tension et le couple ou l'angle de rotation de la vis. Il suffit donc de connaître le couple ou l'angle parcouru durant la montée en couple pour estimer la tension. L'angle donne une meilleure estimation de la tension car il élimine l'effet du frottement lors du vissage (contrairement au couple). Ainsi, l'angle est généralement mesuré à partir du seuil de comptage angulaire fixé par le technicien méthode. Une méthode plus précise (zéro torque-angle) peut être appliquée pour estimer la vraie valeur de l'angle de rotation. La référence [149] est laissée pour décrire en profondeur ce sujet. Pour certaines applications critiques en termes de sécurité (par exemple, les freins de voiture), la vis doit être serrée au-delà de la zone élastique de montée en couple jusqu'à la zone *plastique*, au cours de laquelle se produit une déformation plastique de la vis. La compréhension de la mécanique et des modes de défaillance du procédé permet de garantir un vissage réussi.

### 6.1.2 Détection de défauts et surveillance

Les systèmes de vissage nécessitent une détection et une traçabilité fiables des défauts. En effet, de nombreuses défaillances de produit sont dues à un problème lors de l'assemblage. Même les processus de vissages bien conçus peuvent être perturbés par des problèmes de tolérance des pièces, un mauvais matériau, une lubrification insuffisante, des stratégies de serrage inappropriées ou l'usure des outils. La figure 6.3 montre quelques cas de défaillance typiques [79] pour un vissage. De ce fait, la détection des défauts est nécessaire pour assurer et améliorer la qualité et la productivité.

L'une des méthodes les plus couramment utilisées est la méthode par apprentissage "teach". Cette méthode consiste à définir une courbe couple-angle de référence. Cette dernière est obtenue en moyennant les courbes associées à des serrages correctes. Puis, un écart important entre une nouvelle courbe et la celle de référence indique la présence potentielle d'un défaut. Cette méthode est relativement facile à mettre en œuvre, mais elle manque de flexibilité et de généralité. En effet, elle demande une grande quantité de données afin de définir la courbe de référence, à partir de nombreux essais à réaliser. De plus, elle exige l'utilisation des courbes associées aux serrages corrects, ce qui est difficilement réalisable dans une production en masse. Enfin, il est difficile de fixer le seuil de détection d'un défaut potentiel. Par conséquent, d'autres méthodes ont été proposées.

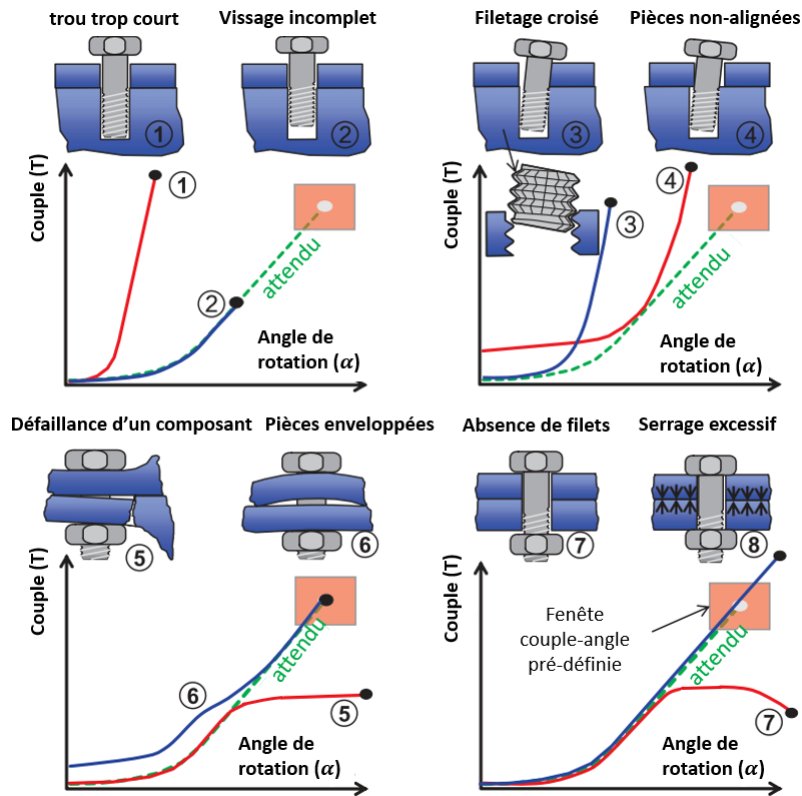


FIGURE 6.3 – Courbe couple-angle durant l'opération de vissage [79]

Ces dernières se répartissent en deux catégories : basée sur des modèles ou basées sur les données.

Les méthodes basées sur des modèles [77, 32] nécessitent des modèles analytiques et des paramètres précis du système. Les approches basées sur des modèles sont flexibles, mais les modèles précis du système sont difficiles à établir et à recalibrer. Les méthodes tirées par les données, telles que les réseaux de neurones classiques [4], les machines à vecteurs de support [104, 127], les réseaux neuronaux convolutifs (CNN) [113], LSTM [22] et les auto-encodeurs [137] apprennent directement à partir de données étiquetées, en supervisé donc. Les méthodes tirées par les données ne nécessitent pas de connaissances préalables, mais elles exigent une grande quantité de données.

La plupart des méthodes mentionnées ci-dessus peuvent atteindre une précision d'environ 90 % ou plus dans la détection de défauts. Cependant, de nombreux points doivent être améliorés avant d'être déployés sur les chaînes de production. Une fiabilité très proche de 100% est nécessaire sur application industrielle, 10% de détections erronées sont inacceptables. De plus, une limite générale est que la connaissance du domaine est rarement

intégrée dans les approches. En outre, les ensembles de données d'entraînement sont généralement issus d'expériences réalisées dans des conditions de laboratoire idéales, où quelqu'un peut manuellement étiqueter chacune des quelques opérations comme correcte ou incorrecte. A l'inverse, connaître la vérité du terrain dans un contexte industriel, avec une production intensive, est presque impossible. C'est la raison pour laquelle les méthodes d'apprentissage non-supervisées sont nécessaires. De plus, même si des étiquettes pouvaient être fournies, ces méthodes sont généralement entraînées sur un nombre limité de défauts connus. Lorsqu'un procédé a été étudié en profondeur, ses principales défaillances sont bien connues et peuvent être surveillées avec succès. Mais pour les procédés de fabrication moins matures, la détection de défauts inconnus est un problème important.

## 6.2 Découverte incrémentale de nouveaux défauts

Dans cette partie, nous allons nous baser sur l'architecture proposée dans le chapitre 4 pour effectuer de la détection incrémentale de défauts sur des courbes de vissage.

### 6.2.1 Protocole expérimental

**Jeu de données** Le cas d'étude porte sur le vissage d'injecteur de moteur de voiture. 4096 opérations de vissage ont été effectuées pendant cinq jours de production d'une visseuse automatisée. Les séries temporelles de couple et de position angulaire ont été mesurées. Avec 4096 observations, déterminer quelle opération était correcte ou non est impossible manuellement. Par conséquent, une approche non-supervisée, telle que proposée, est appropriée. Les 2000 premières opérations ont constitué le jeu de données d'entraînement (peuplement de l'ontologie). Le second jeu de données a été utilisé pour les classifications en temps réel avec option de rejet, avec un traitement périodique des rejets (permettant l'enrichissement de l'ontologie).

**Pré-traitement** Des connaissances empiriques sont intégrées lors du choix des l'extraction de caractéristiques (feature extraction). La phase de montée en couple, connue comme critique, est ainsi sélectionnée. Dans les séries temporelles, la fenêtre temporelle correspondante est automatiquement détectée entre le seuil de comptage angulaire  $T_0$  (fi-



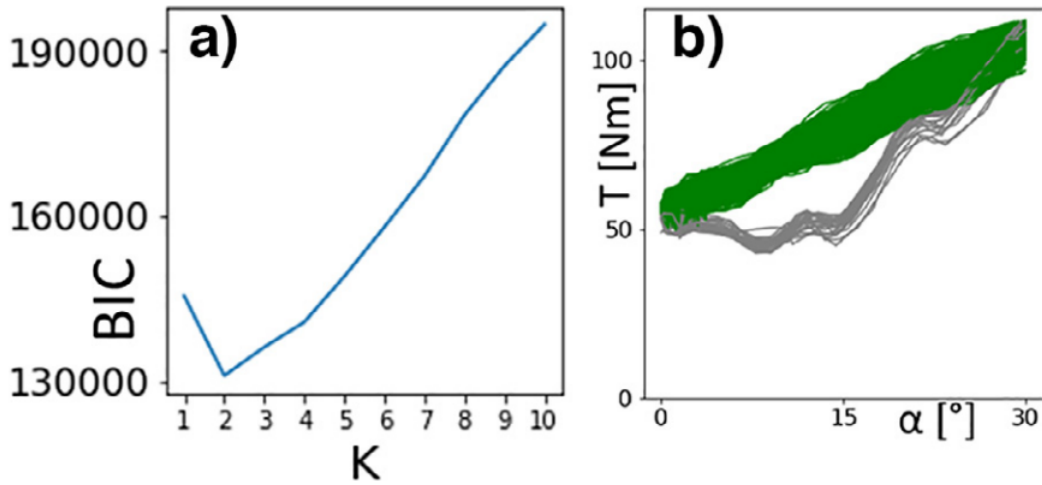


FIGURE 6.4 – Peuplement de l'ontologie : a) Score du BIC, b) Visualisation des ( $K^*$ ) clusters.

gure 6.2) et le couple maximum. Une fois les signaux de couple transposés dans le domaine angulaire, des caractéristiques sont extraites pour chaque serrage. Un ensemble de 9 caractéristiques appropriées pour la surveillance du vissage, inspirées de [32], a été choisi : la moyenne, l'écart-type et l'étendue du signal et de ses dérivées premières et secondes. Les autres paramètres de l'expérience ont été fixés de même manière que dans les expériences présentées dans la section 4.2.1. En effet, le nombre maximal de cluster  $K_{max}$  a été fixé à 50 et  $\tau$  à 1% pour le test de rejet (équation 4.1). Le type de paramètres de la matrice de covariance à utiliser a été fixé à "complet".

## 6.2.2 Peuplement de l'ontologie

Comme le montre la figure 6.4.a, un nombre optimal ( $K^*$ ) de deux clusters a été trouvé par BIC (équation 2.22) et en appliquant l'apprentissage non-supervisé sur le jeu de données d'entraînement. Une visualisation des signaux de couple de serrage, dans le domaine angulaire, des clusters initiaux a été proposée dans la figure 6.4.b, permettant une analyse par l'expert. Cet expert a déclaré que le cluster vert correspondait à des opérations de vissage correctes et a été étiqueté comme "OK". Il a également déterminé que le deuxième cluster, en gris, correspondait à un problème de "Stick & Slip" et a été étiqueté " $Stick_1$ ". De cette façon,  $N_{OK} = 1975$  et  $N_{Stick_1} = 25$  opérations correctes et défectueuses ont été respectivement trouvées dans l'ensemble de données d'entraînement. Ces États de vissage "OK" et de défaut de stick&slip " $Stick_1$ " ont été introduits dans l'ontologie de détection de

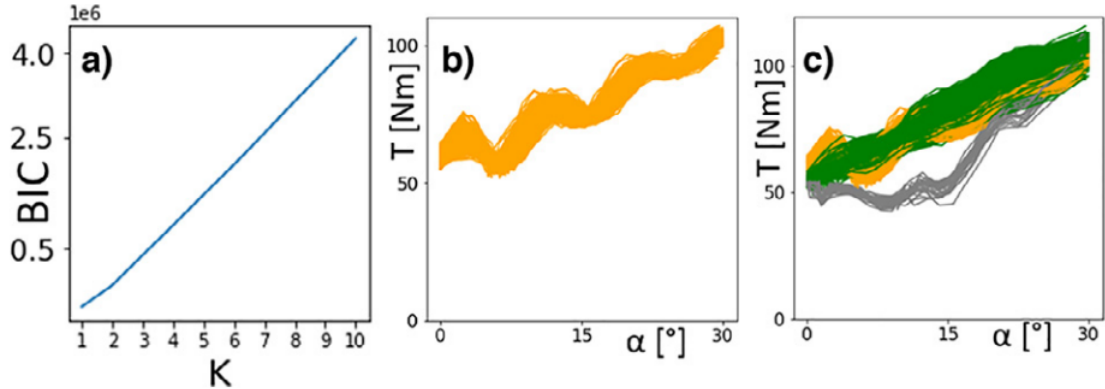


FIGURE 6.5 – Enrichissement de l’ontologie : a) Score du BIC sur le jeu d’exceptions, b) Cluster appris sur le jeu d’exceptions c) Clusters finaux

défauts (ODD), avec leurs Signatures et nombre d’Occurrences respectives. Notons qu’ici chaque Signature se compose d’une seule gaussienne (mais il aurait pu s’agir de plusieurs composantes). Le Contexte correspond au modèle de machine et aux paramètres opératoires utilisés pour ces vissages. Cette étape a ainsi permis d’initier le peuplement de l’ODD à partir du jeu de données d’entraînement. Désormais, la classification en temps réel et l’enrichissement de l’ontologie peuvent être entamées avec l’ontologie ainsi peuplée.

### 6.2.3 Classification et enrichissement

La classification en temps réel avec l’option de rejet a ensuite été réalisée sur le deuxième ensemble de données de test (2096 vissages). Comme observable sur la figure 6.5.c, le modèle de surveillance a déterminé que 1622 opérations de vissage étaient correctes (dans le cluster vert "OK") et 29 étaient des défauts "Stick1" (en gris). Par ailleurs, 445 opérations ont été rejetées (en orange). En traitant l’ensemble des exceptions, le score BIC a déterminé qu’un nouveau défaut était apparu (figure 6.5.a). En visualisant la figure 6.5.b, l’expert a confirmé qu’il s’agissait d’un deuxième type de problème de stick & slip, plus difficile à détecter. Le nouveau cluster a été étiqueté "Stick2" et l’ontologie a été enrichie avec sa Signature associée.

La chronologie des opérations de vissage après classification est illustrée dans la figure 6.6. Cela permet une traçabilité de la production industrielle. La ligne noire verticale sépare la phase de peuplement initial, de la phase d’enrichissement. On peut observer que le second défaut, en orange, n’apparaît que le dernier jour, alors que le premier défaut,

en gris, semble aléatoire.

Finalement, 3597 "OK", 54 "Stick1" et 445 "Stick2" ont été trouvés et un nouveau défaut a été découvert. Les classes étaient déséquilibrées, ce qui est plus difficile pour l'apprentissage automatique non-supervisé. En outre, l'expert a validé par l'exploration manuelle des données qu'il n'y avait pas d'erreur de classification. Par conséquent, l'approche proposée a été considérée comme validée. Contrairement à une approche purement orientée données, elle s'appuie sur des les interventions ponctuelles d'un expert qui intègre des connaissances empiriques, ce qui apporte un haut niveau de confiance dans le processus d'exploration de données. Cela rend la solution proposée plus fiable dans un contexte industriel. En outre, il est beaucoup plus pertinent de solliciter l'expert pour labéliser quelques clusters, que pour labéliser des milliers d'observations de vissage ou autre.

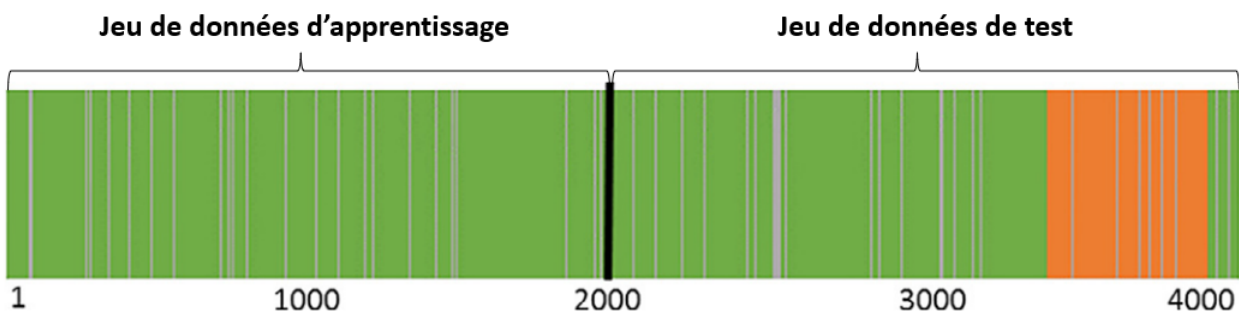


FIGURE 6.6 – Traçabilité des opérations de vissage classifiées par monitoring avec l'I.A. proposée

## 6.2.4 Conclusion

L'étude de cas concernait la surveillance du vissage automatisé dans l'industrie automobile. Les caractéristiques ont été extraites des signaux de couple dans le domaine angulaire pendant la phase de montée en couple. Un cluster correct et un cluster défectueux ont été trouvés dans le jeu de données d'entraînement et étiquetés par l'expert, ce qui a alors permis d'initier le peuplement de l'ontologie de détection de défaut (ODD). Ensuite, un nouveau défaut a été découvert dans un second jeu de données, enrichissant progressivement l'ontologie et validant ainsi l'approche proposée.

TABLE 6.1 – Nombre d’instances par classe et par Contexte dans chaque jeu de données

Jeux de données	Contexte 0	Contexte 1	Contexte 2	Contexte 3	Contexte 4
1 <sup>er</sup> jdd	2055	678	11239	1684	-
2 <sup>ème</sup> jdd	1078-7	451-5	2470-6	6630-56	631-11
3 <sup>ème</sup> jdd	1078-1078	451-451	2470-2470	6630-6630	631-631

## 6.3 Généralisation des connaissances

Dans cette partie, les mécanismes de généralisation de connaissances définis dans le chapitre 5 seront appliqués dans le domaine du vissage.

### 6.3.1 Protocole expérimental

**Définition des contextes** Dans le vissage, nous considérons un Contexte comme une opération de vissage qui a été effectuée avec un outil donné (visseuse), suivant une stratégie de vissage (et ses paramètres opératoires) et une application bien spécifique. En d’autres termes, un Contexte une combinaison spécifique de trois attributs {application, stratégie, machine}. Ainsi, on obtient autant de contextes que de combinaisons distinctes de ces 3 attributs.

Enfin, nous considérons des contextes similaires mais différents : ceux qui partagent la même application et la même stratégie de vissage mais différentes visseuses. Par exemple, nous aurons 5 contextes différents et similaires dans le cas suivant : {application, stratégie, machine<sub>*i*</sub>},  $i \in \{1, \dots, 5\}$ .

**Jeux de données** À partir de la définition de contextes ci-dessus. Nous avons récolté deux autres jeux de données à partir d’une base de données provenant de serrage de composants de moteurs. Ces derniers sont différents du jeu de données utilisé dans la section précédente (cf. section 6.2). 4 contextes (i.e. 4 visseuses) sont récoltés sur le 1<sup>er</sup> jeu pour la même application et stratégie, et 5 contextes (i.e. 5 visseuses) sur le 2<sup>ème</sup> (pour la même application et stratégie). Le nombre de classe est de 1 pour le 1<sup>er</sup> jeu de données et de 2 pour le 2<sup>ème</sup>, et ce pour tous les contextes. Il convient de souligner que les étiquettes des classes (État) sont identiques pour tous les contextes sur les 2 jeux de données. En d’autres termes, on retrouve les mêmes défauts sur tous les contextes. Le

nombre d'instances par classe et par contexte est décrit dans le tableau 6.1.

La labélisation des jeux de données a été effectuée par l'expert à travers une IHM illustrant les clusters appris en non-supervisé en amont, conformément à l'approche proposée. De plus, comme le 2<sup>ème</sup> jeu de donnée est très fortement déséquilibré, nous proposons un 3<sup>ème</sup> jeu de donnée augmentant la classe minoritaire. L'augmentation de données pour cette dernière revient à simuler des courbes de vissage du même profil que la classe minoritaire avec une légère déviation aléatoire dans l'enveloppe des courbes. Des illustrations des courbes sont fournies en annexe 6.5. Le pré-traitement des courbes est le même que celui dans la section 6.2.1. Les autres paramètres restent les mêmes que ceux fixés dans la section 5.3.

### 6.3.2 Réduction intra-contextes

Dans cette partie, nous allons appliquer la réduction intra-contextes durant l'apprentissage (section 5.3) pour être réaliste avec l'application industrielle. Comme le 2<sup>ème</sup> jeu de donnée est très déséquilibré, nous ne pouvons pas définir des signatures associées à la classe minoritaire durant l'apprentissage itératif. En effet, chaque jeu de données est divisé en un jeu d'entraînement (80%) et de test (20%), le nombre d'observations de la classe minoritaire sera inférieur au nombre des caractéristiques fixé (i.e. 9), sauf pour le contexte 3. De ce fait, les gaussiennes qui seront éventuellement apprises sur les jeux de données d'entraînement ne seront statistiquement pas représentatives de la classe minoritaire, et de ce fait les métriques résultantes sur la classe minoritaire ne seront pas fiables. Par conséquent, seules les observations de la classe majoritaire seront considérées dans l'expérience itérative. Ainsi, nous ne mesurons pas les taux de reconnaissance sur le 1<sup>er</sup> et 2<sup>ème</sup> jeu de donnée sur les scénarios itératifs. Ce sera néanmoins suffisant pour démontrer les propriétés de la méthode proposée. Pour le 3<sup>ème</sup> jeu de donnée, nous simulons l'arrivée des instances de défauts de façon aléatoire.

On observe sur la figure 6.7 que les taux de reconnaissabilité sont largement meilleurs lorsque la réduction est appliquée. De ce fait, l'architecture permet de reconnaître la quasi-totalité des instances au bout de quelques étapes d'enrichissement. De plus, le nombre de ces étapes est largement réduit pour le contexte 1. En outre, ces très bons résultats sont obtenus avec un nombre de gaussiennes très faible (cf. figure 6.7.b). Ceci démontre à nouveau l'intérêt de la généralisation des Signatures. Le même constat est réalisé pour

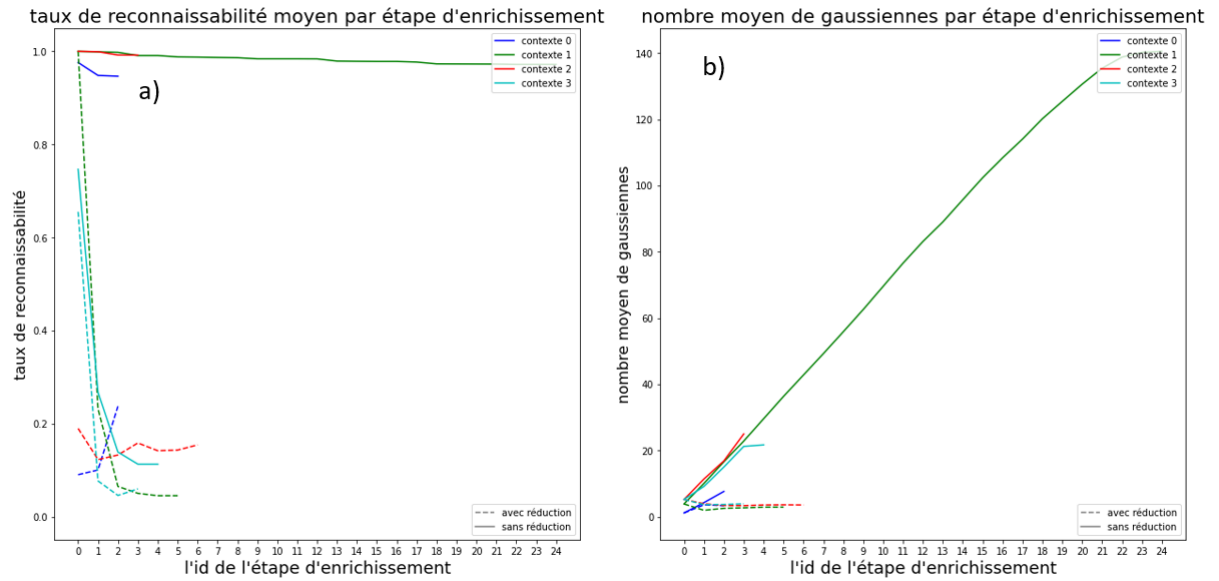


FIGURE 6.7 – Résultats de réduction intra-contexte sur le 1<sup>er</sup> jeu de données : a) taux de reconnaissance, b) nombre moyen de gaussiennes

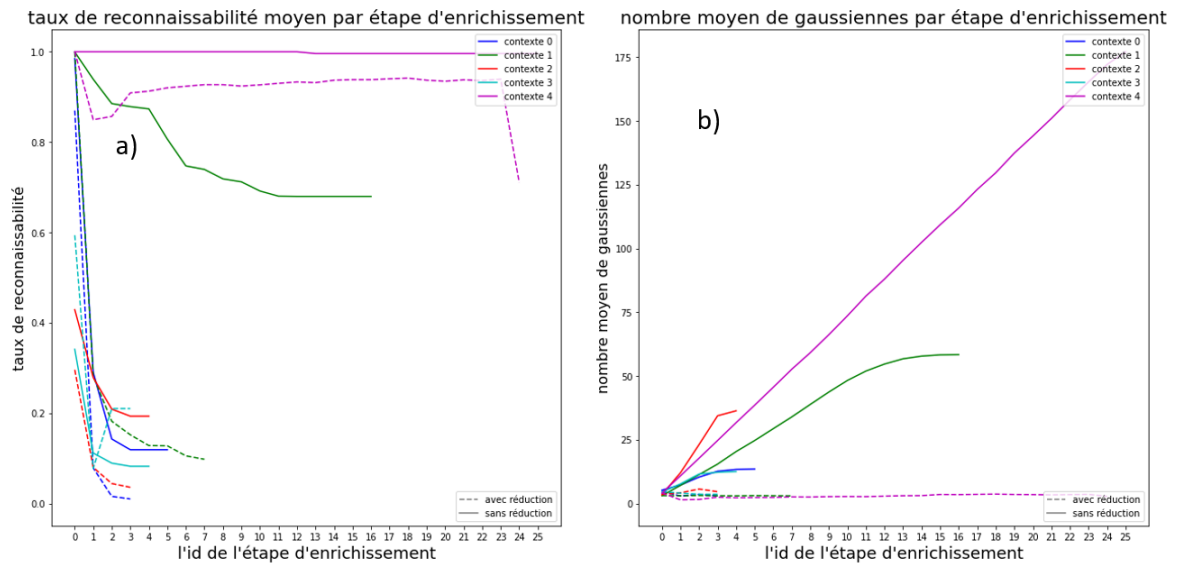


FIGURE 6.8 – Résultats de réduction intra-contexte sur le 2<sup>ème</sup> jeu de données : a) taux de reconnaissance, b) nombre moyen de gaussiennes

le 2<sup>ème</sup> jeu de données (figure 6.8).

Par ailleurs, nous observons une réduction significative du nombre de gaussiennes dans le 3<sup>ème</sup> jeu de données (figure 6.9.c). Cette réduction a permis d'améliorer considérablement les métriques pour le contexte 1. De plus, même si le taux de reconnaissance est

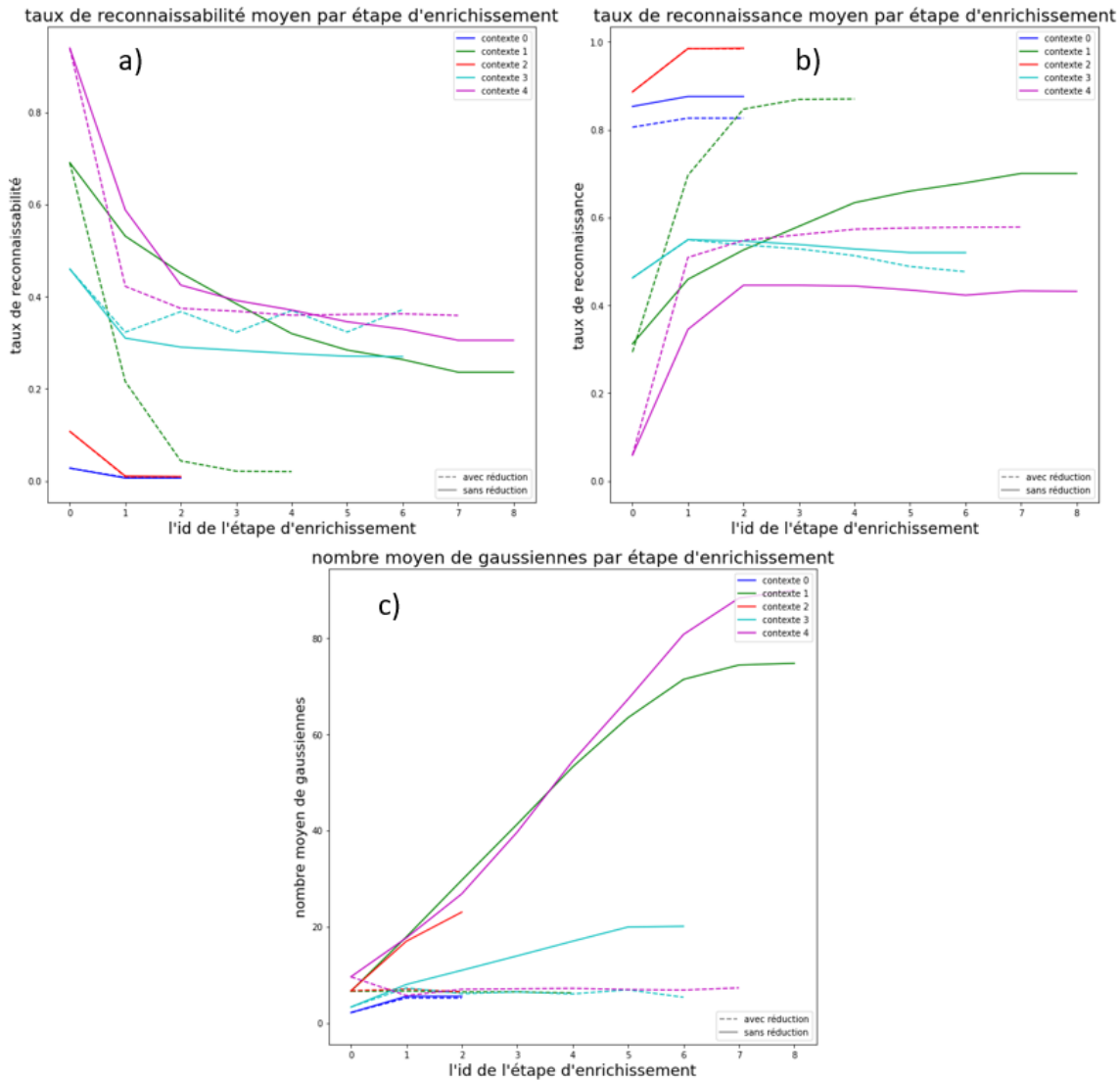


FIGURE 6.9 – Résultats de réduction intra-contexte sur le 3<sup>ème</sup> jeu de données : a) taux de reconnaissabilité, b) taux de reconnaissance, c) nombre moyen de gaussiennes

inférieur à la fin de l'apprentissage (figure 6.9.a), le taux de reconnaissance quant à lui est nettement meilleur pour le contexte 4 (figure 6.9.b). La solution reconnaît moins d'observations à la fin de l'expérience, mais la qualité d'affectation des observations reconnues est nettement supérieure. En revanche, nous remarquons une diminution mineure de ces dernières pour les contextes 3 et 0. Cette dernière est expliquée par le fait que la réduction a éliminé une gaussienne pertinente pour le contexte 0 (figure 6.9.c), ce qui a diminué le taux de reconnaissance. Pour le contexte 3, le problème est vraisemblablement lié à la variabilité inerte des courbes et leur labélisation. En effet, la labélisation s'est faite de

manière subjective dans le but d'avoir les mêmes classes de défauts identifiées dans les autres contextes. En revanche, nous pouvons visualiser à travers les figures de l'annexe (cf. 6.5, A.10) que certaines courbes de la classe minoritaire sont très similaires à celles de la classe majoritaire (sauf dans les valeurs finales du couple). De plus, des courbes n'appartenant à aucune des deux classes ont été mixées avec la classe majoritaire. En réalité, le clustering par BIC a donné un nombre de cluster égal à 7. De ce fait, la multitude des classes a engendré une baisse des taux de reconnaissabilité et de reconnaissance moyens. Nous pouvons donc affirmer qu'une mauvaise labélisation, ou une labélisation forcée, peut engendrer une baisse des performances. Enfin, la réduction intra-contextes a réduit significativement le nombre des gaussiennes, mais cela n'a eu aucun effet sur les taux de reconnaissabilité et de reconnaissance sur le contexte 2.

Suite à ces résultats, nous pouvons affirmer que la réduction intra-contextes permet d'améliorer la détection de défauts dans le vissage dans le cas où la labélisation par l'expert est correctement effectuée.

### 6.3.3 Généralisation inter-contextes

Nous allons appliquer la généralisation ponctuelle sur les 3 jeux de données. En revanche, comme pour la partie précédente, les observations de la classe minoritaire ne sont pas retenues pour le 2<sup>ème</sup> jeu de données dans le cadre de la généralisation appliquée à l'apprentissage itératif.

**Généralisation ponctuelle** La généralisation ponctuelle inter-contextes (cf. section 5.3.3) consiste à étudier l'intérêt qu'aura cette dernière, lorsque des signatures issues de contextes connus seront généralisées. L'objectif est d'étudier si des signatures généralisées permettront d'améliorer la détection de défauts dans des contextes inconnus, en les comparant à une simple agrégation de signatures. De plus, il convient d'étudier si la performance s'accroît en fonction du nombre de contextes pris. De ce fait, la généralisation ponctuelle commence par apprendre un GMM en supervisé dans chaque classe de défauts dans des contextes supposés connus. Puis, des gaussiennes généralisées par le mécanisme de généralisation inter-contextes seront obtenues pour chaque classe de défauts. Enfin, les gaussiennes généralisées serviront à la classifier des observations des contextes supposés inconnus. La généralisation s'effectue en prenant progressivement plus de contextes lors de l'apprentissage.



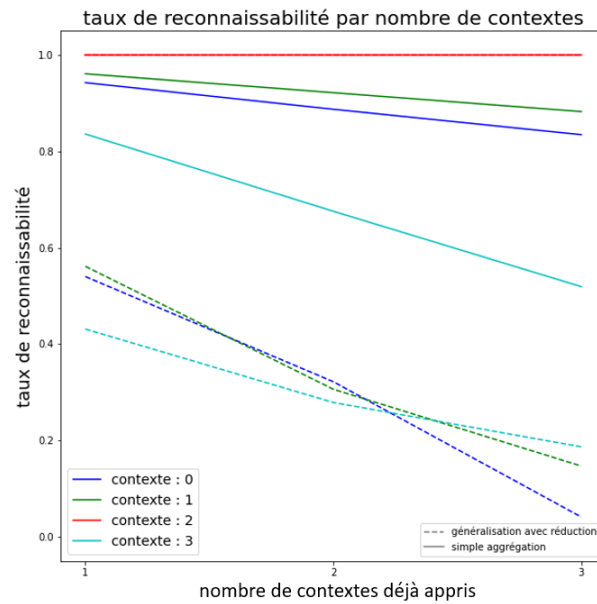


FIGURE 6.10 – Résultats de la généralisation ponctuelle inter-contextes sur le 1<sup>er</sup> jeu de données : taux de reconnaissabilité

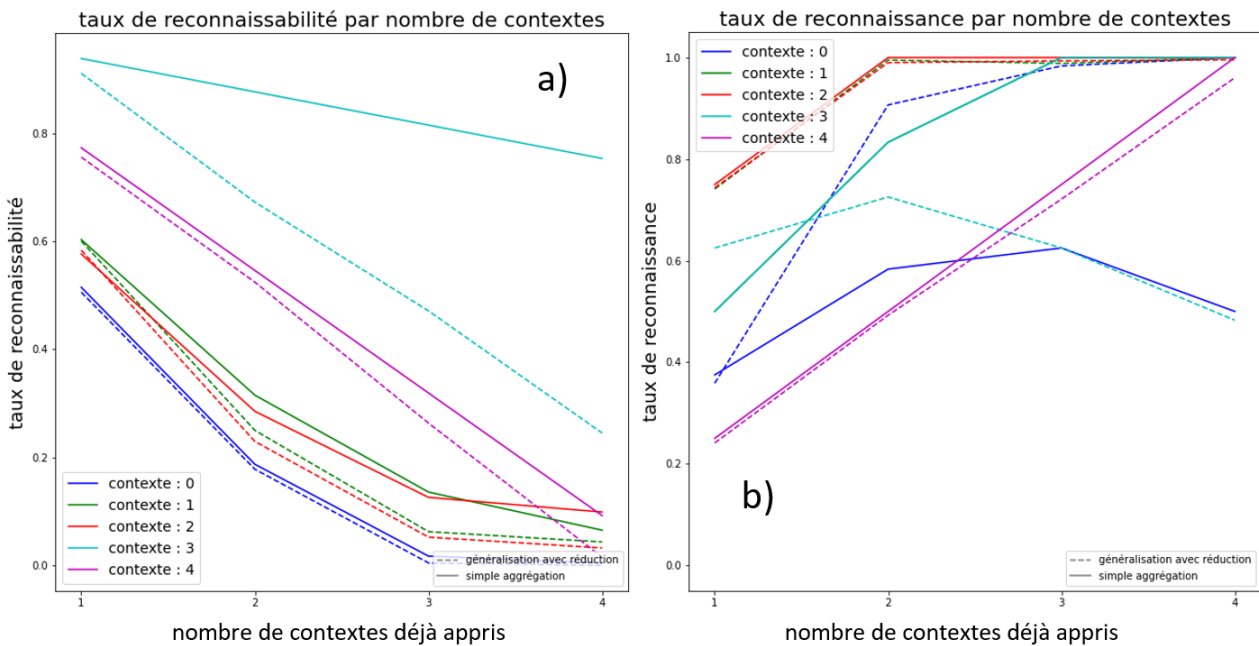


FIGURE 6.11 – Résultats de la généralisation ponctuelle inter-contextes sur le 2<sup>ème</sup> jeu de données : a) Taux de reconnaissabilité, b) Taux de reconnaissance

On observe à travers la figure 6.10 que les taux de reconnaissabilité sont meilleurs lorsque la généralisation ponctuelle inter-contextes est effectuée. En effet, la détection des

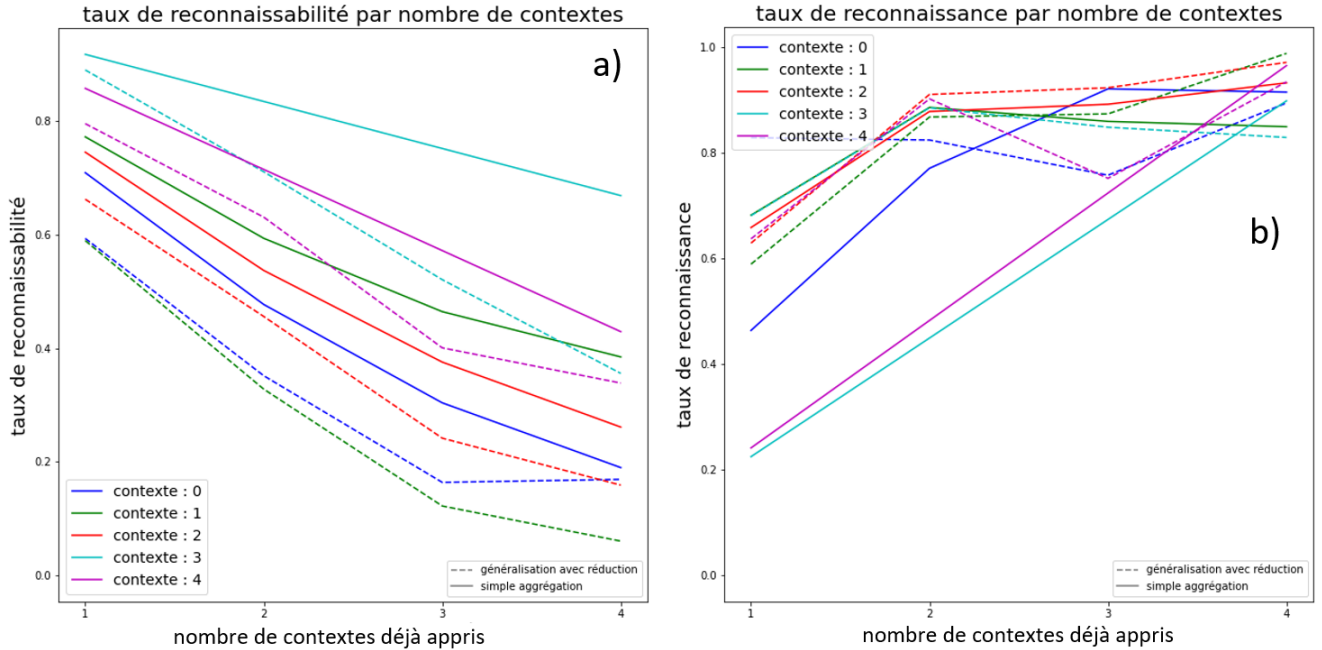


FIGURE 6.12 – Résultats de la généralisation ponctuelle inter-contextes sur le 3<sup>ème</sup> jeu de données : a) Taux de reconnaissabilité, b) Taux de reconnaissance

observations dans le contexte inconnu s'améliore au fur à mesure que plus de contextes sont pris lors de la généralisation inter-contextes, atteignant des valeurs en dessous de 10% pour le contexte 0. En revanche, nous observons que la généralisation n'apporte pas d'amélioration dans le contexte 2 même en généralisant sur tous les autres contextes. Ce qui nous indique le contexte 2 est semble t'il éloigné des autres contextes.

Pour le 2<sup>ème</sup> jeu de donnée, nous remarquons que la généralisation améliore le taux de reconnaissabilité (figure 6.11.a), en particulier pour le contexte 3. En parallèle, nous observons que le taux de reconnaissance est presque identique pour les contextes 1, 2, 4 (figure 6.11.b). Ce qui nous indique que la qualité de détection de défauts (taux de reconnaissance) n'a pas été détériorée par l'amélioration de la capacité de détection de défauts (taux de reconnaissabilité). En d'autres termes, la solution détecte plus de défauts et garde les mêmes précisions. Par ailleurs, nous observons une nette amélioration pour le contexte 0 en termes du taux de reconnaissance avec un taux de reconnaissabilité presque identique. À l'inverse, une détérioration pour le contexte 3 est constatée par rapport au taux de reconnaissance. Par conséquent, même si la généralisation améliore la détection des observations dans le contextes inconnus, l'affectation dans leurs classes respectives n'est pas toujours assurée. Ce phénomène est probablement lié à au déséquilibre des

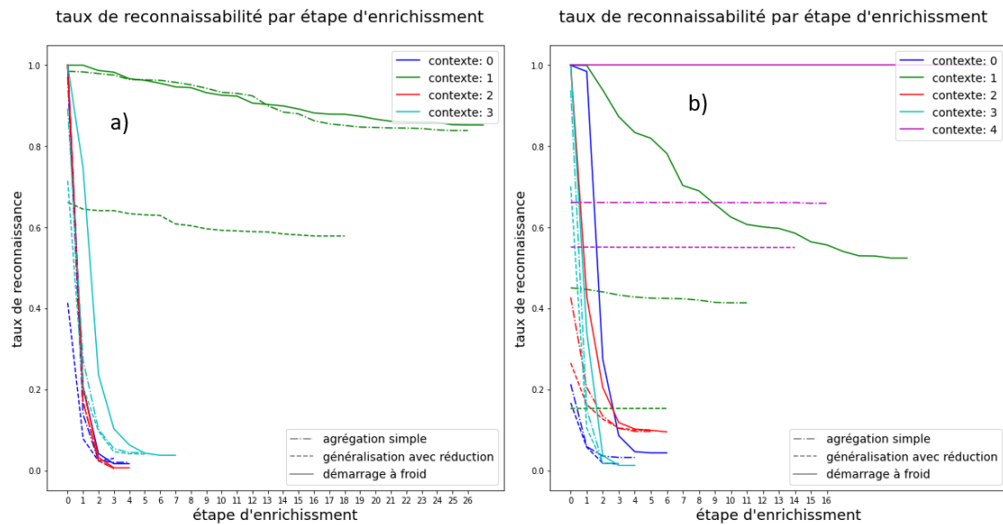


FIGURE 6.13 – Résultats de la généralisation inter-contextes appliquée à l'apprentissage itératif : a) taux de reconnaissance sur le 1<sup>er</sup> jeu de donnée, b) taux de reconnaissance sur le 2<sup>ème</sup> jeu de donnée

classes et à la labélisation subjective sur le contexte 3.

En effet, en appliquant la généralisation sur le 2<sup>ème</sup> jeu de données équilibré (i.e, 3<sup>ème</sup> jeu de données), nous observons sur la figure 6.12 que les taux de reconnaissance sont nettement meilleurs lorsque la généralisation est appliquée. De plus, ces derniers s'améliorent en fonction du nombre de contexte pris pour la généralisation. Enfin, nous remarquons une légère amélioration des taux de reconnaissance (figure 6.12.b) (sauf pour le contexte 3, probablement à cause de la labélisation subjective). Globalement, il est clair que la généralisation de signatures est meilleure qu'une simple agrégation de signatures. En effet, la généralisation inter-contextes a permis d'améliorer la détection des observations dans des contextes inconnus tout en assurant une qualité d'affectation légèrement meilleure. Enfin, il convient de souligner que les performances de la généralisation s'améliorent en fonction du nombre de contextes connus pris dans l'apprentissage.

**Généralisation inter-contextes appliquée à l'apprentissage itératif** La généralisation inter-contextes appliquée à l'apprentissage itératif consiste dans un premier temps à apprendre des signatures issues de plusieurs contextes supposés connus de façon itérative (avec réduction) selon le schéma décrit dans la section 4.2.3, et de les capturer par l'ontologie. Ensuite, une simulation d'un arrivage en temps réel d'observations issues de contextes inconnus est déclenchée, toujours selon le schéma décrit dans la section 4.2.3.

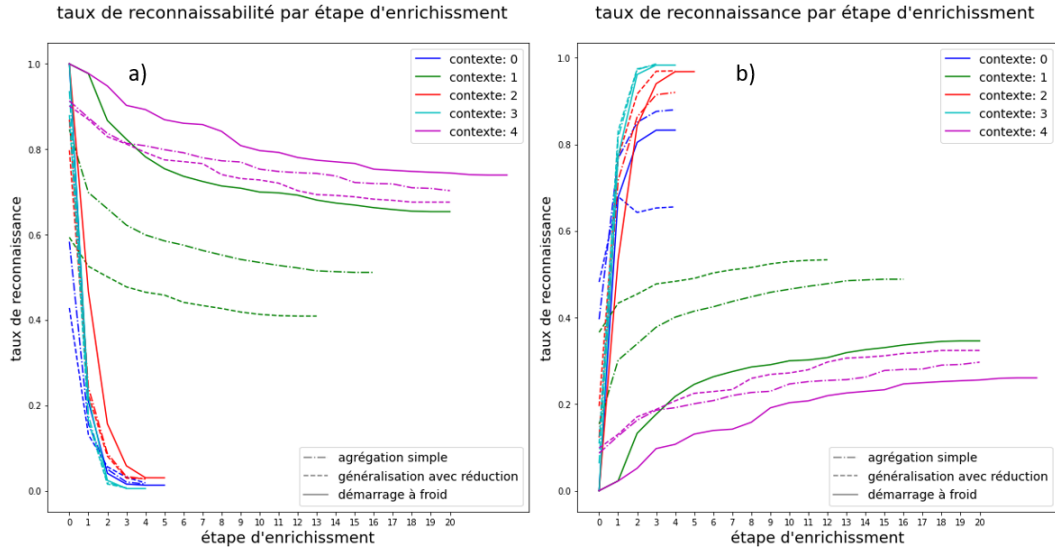


FIGURE 6.14 – Résultats de la généralisation inter-contextes appliquée à l'apprentissage itératif sur le 3<sup>ème</sup> jeu de données : a) Taux de reconnaissabilité, b) Taux de reconnaissance

L'objectif est d'étudier la performance de 3 méthodes de traitement de ces observations (démarrage à froid, généralisation par agrégation, généralisation multi-contextes). La section 5.3.3 fournit la description complète de l'expérience.

Nous observons à travers la figure 6.13.a que la généralisation itérative inter-contextes améliore le taux de reconnaissabilité pour le contexte 1 durant tout l'apprentissage. Par contre, l'agrégation simple des signatures des autres contextes déjà appris est équivalente au démarrage à froid, où aucune signature n'a été apprise au début de l'expérience. Pour les autres contextes, nous observons que les métriques convergent vers le même point final. En revanche, nous remarquons que les taux sont nettement meilleurs au début de l'expérience, ce qui nous indique que l'architecture exploite bénéfiquement les connaissances apprises dans des contextes similaires pour améliorer la détection des observations dans le contexte inconnu : en détectant mieux plus tôt. Ce comportement est fortement avantageux dans le cas où la fréquence d'arrivage des observations est très faible. Le même constat est réalisé pour le 2<sup>ème</sup> jeu de donnée (figure 6.13.b) avec une nette amélioration pour les contextes 1 et 4.

Concernant le 3<sup>ème</sup> jeu de données (figure 6.14), nous observons aussi une nette amélioration des métriques durant toute l'expérience, pour les contextes 1 et 4 lorsque la généralisation est utilisée. Pour les autres contextes, les métriques sont meilleures au début de l'expérience, et ensuite convergent vers le même point finale (sauf pour le contexte

0).

Globalement, l'utilisation des signatures généralisées issues de contextes similaires nous permet de résumer les connaissances apprises par ailleurs pour ensuite être transférées et utilisées dans des contextes inconnus. Cette opération nous permet d'améliorer clairement la reconnaissance des défauts déjà détectés dans des contextes similaires et de ne pas attendre un nombre d'observations suffisant pour apprendre des signatures. Cela nous donne un avantage considérable dans le cas où la fréquence d'arrivage des données est très faible et/ou que la probabilité d'apparition des défauts rencontrés par ailleurs est très rare ; ce qui est généralement le cas sur production industrielle.

### **6.3.4 Conclusion**

Nous avons montré à travers cette partie que la généralisation a permis d'améliorer la détection de défauts dans le vissage. En effet, la réduction intra-contexte a permis de réduire le sur-apprentissage et les gaussiennes obsolètes, ce qui a fortement augmenté les métriques associées à la détection de défauts.

La généralisation inter-contextes a nettement augmenté la qualité de détection de défauts dans des contextes inconnus. En effet, elle permet d'exploiter les connaissances apprises et généralisées dans contextes similaires. De plus, les performances s'améliorent au fur à mesure qu'on introduit plus de contextes.

## **6.4 Industrialisation de l'architecture**

La première version de notre architecture (chapitre 4 et section 6.2) a fait l'objet d'un dépôt de brevet européen. En outre, elle a été validée par simulation sur plusieurs jeux de données industriels. Enfin, des améliorations incrémentales ont été proposées sur ses différentes briques fonctionnelles (ex : amélioration du BIC, ajustement des caractéristiques).

En parallèle, une application informatique intégrée dans le logiciel DeMeter de Desoutter a été développée. Ce développement a été effectué durant ma deuxième et troisième année de thèse. De ce fait, Desoutter est capable de proposer une solution qui permet d'effectuer du monitoring en temps réel par Intelligence Artificielle sur les opérations de vissage. Elle permet de détecter de nouveaux défauts durant la production, tout en interagissant avec l'expert du domaine. Cette partie pratique m'a permis de prendre

connaissance des contraintes industrielles de la mise en production, telles que : la difficulté de communication entre les différentes équipes, la maintenabilité de la solution logicielle et les contraintes des différentes technologies utilisées pour l'implémentation industrielle. Actuellement, un essai sur le terrain est en train d'être organisé pour tester la solution en temps réel chez un client du secteur automobile.

En revanche, des difficultés liées aux jeux de données limitent l'utilisation de l'architecture : nombre de points par courbe de vissage, contextes différents mais sans attribut pour les distinguer dans la base de données (changement de vis par exemple). De ce fait, même si elle l'architecture a été validée sur plusieurs jeux de données, il existe quelques contraintes techniques non négligeables qui limitent actuellement l'exploitation de la solution.

De plus, la partie concernant la généralisation n'a pas encore été implémentée dans l'application. Deux raisons expliquent cela. La première est que la première version (chapitre 4) n'a pas encore été totalement validée sur le terrain. La deuxième est qu'il est difficile de la faire valider sur plusieurs jeux de données industriels. En effet, cette validation est fortement limitée par la disponibilité des données et des contextes similaires. De plus, il faudra qu'un défaut soit appris sur plusieurs contextes (au moins deux) simultanément, ce qui est rarement le cas en industrie. Nous avons montré à travers les 2 derniers chapitres l'intérêt que peut apporter une telle solution. Par conséquent, nous pouvons envisager à court terme une première implémentation de la généralisation une fois que les deux problèmes auront été résolus.

## 6.5 Conclusion

Dans ce chapitre, nous avons commencé par appliquer la détection de défaut inconnu dans le domaine du vissage, à travers l'architecture ML-KM formalisée dans le chapitre 4. Cette dernière permet d'apprendre et de capturer les signatures de défauts inconnus, issues de plusieurs contextes, dans une ontologie à travers les résultats obtenus par apprentissage automatique. La labélisation et l'interprétation des résultats est soutenue par les connaissances de l'expert. De plus, la généralisation des connaissances proposée dans le chapitre 5 (relatives aux signatures, intra et inter contextes) a été également testée. Les données récoltées proviennent d'une base de données client du secteur automobile. La présentation du métier suivi d'une petite revue de littérature sur la détection de défauts de vissage a

été présentée dans un premier temps. Ensuite, l'application du mécanisme de détection de défauts inconnus a été testé et validé. Enfin, les 2 mécanismes de généralisation de signatures ont été appliqués.

L'application de la détection incrémentale de défauts a permis de montrer, une fois encore, l'intérêt de l'approche proposée à détecter itérativement des défauts inconnus qui n'étaient pas présents au début de la surveillance process. De ce fait, l'architecture proposée est sensible à l'apparition de nouveaux événements (i.e. nouveaux défauts) qui peuvent survenir pendant le procédé de fabrication, ce qui lui donne un avantage important par rapport aux méthodes de ML conventionnelles, qui se limitent aux défauts connus seulement. Par ailleurs, le premier mécanisme de généralisation des connaissances a permis d'améliorer la détection de défauts par la réduction du sur-apprentissage. Le deuxième mécanisme a montré l'intérêt d'exploiter les connaissances dans des contextes similaires pour être exploités dans de nouveaux contextes inconnus. Globalement, les deux mécanismes ont amélioré la détection de défauts dans le domaine du vissage malgré les imperfections liées aux données (nombre de points dans la courbe de vissage, cohérence des défauts sur les contextes et labélisation très subjective).

L'architecture ML-KM proposée a permis d'améliorer considérablement la surveillance des opérations de vissage. En effet, la détection de défauts inconnus a permis de surveiller en temps réel l'apparition de nouveaux comportements qui peuvent nuire à la production, ce qui lui donne un avantage par rapport aux méthodes ML conventionnelles qui se limitent uniquement à des défauts connus. De plus, la découverte et la formalisation des connaissances est soutenue conjointement par l'expert et les méthodes de ML. En effet, contrairement aux approches de KM classiques qui se base sur les connaissances métiers qui peuvent être limitées (qualité d'implication de l'expert, obsolescence des connaissances, etc.), l'architecture KM-ML permet combler ce manque par l'exploitation des méthodes de ML. Cela permet d'être cohérent avec la réalité, et de faciliter la formalisation des connaissances de l'expert en le guidant par les méthodes de ML. Par ailleurs, la gestion de multitude sources de connaissances issues des différents contextes opératoires a été facilitée par une ontologie générique. Contrairement aux approches de ML traditionnelles qui se limitent à un contexte opératoire, la gestion de différents contextes est facilement traitable avec l'architecture ML-KM. Enfin, l'exploitation des connaissances intégrées itérativement dans l'ontologie permet d'améliorer la détection de défauts, qui rappelons est la tâche centrale de notre étude. En effet, nous avons montré que l'exploitation des signatures issues de contextes similaires permet d'améliorer la capacité de détection des

défauts. De plus, nous avons montré que cette capacité augmente au fur et à mesure du nombre de contexte appris. De ce fait, comme l'architecture permet d'exploiter aisément de nombreux contextes simultanément, cela lui donne un avantage conséquent par rapport aux méthodes de l'Adaptation au Domaine qui se limitent souvent à 2 contextes (i.e. domaine source et domaine cible), et utilisent des réseaux de neurones profonds avec des architectures peu génériques et difficilement interprétables.

Enfin, une limitation non négligeable est que la généralisation fonctionne que dans le cas où nous avons les mêmes défauts sur plusieurs contextes. En effet, nous avons montré qu'il faut avoir les mêmes classes de défauts sur tous les contextes afin d'effectuer la généralisation. Dans le cas contraire, nous avons vu que cela peut détériorer la solution (labélisation peu fiable dans le contexte 3 du 2<sup>ème</sup> jeu de données). De plus, nous avons vu qu'un important déséquilibre des classes (2<sup>ème</sup> jeu de donnée, tableau 6.1 affecte négativement la solution. Nous adresserons ces deux limitations dans des travaux futurs.





# CONCLUSION

---

Dans ce travail, nous avons proposé une architecture solide et fiable pour la surveillance des procédés de fabrication en temps réel. Plusieurs problèmes qui découlent de cette tâche ont été abordés comme : la détection de défauts inconnus, la capacité à intégrer la connaissance de défauts nouvellement détectés dans la surveillance, la capacité à gérer plusieurs contextes industriels différents (conditions opératoires, outils, etc), et la capacité à surveiller des contextes inconnus dans lesquels peu de données sont disponibles. Nous avons réussi à atteindre ces objectifs en pratique, en proposant une architecture qui repose sur la combinaison de techniques d'apprentissage automatique (ML) assistées par les connaissances de l'expert, et d'une base de connaissance modélisée par une ontologie. Un nouveau mécanisme de ML permet de détecter des défauts émergeant dans un contexte particulier, de les faire labéliser par un expert du domaine et de les intégrer dans une ontologie sous forme de connaissances. L'ontologie permet d'organiser les connaissances apprises dans chacun des contextes, permettant ainsi une gestion efficace. Ces connaissances serviront dans un premier temps à effectuer de la classification en temps réel sur les données de monitoring d'opérations de fabrication dans des contextes connus. Cette combinaison permet l'améliorer la qualité et la performance de chaque volet. En effet, les techniques de ML permettent d'enrichir l'ontologie avec les connaissances issues de l'exploitation des données de monitoring d'opération et de la connaissance de l'expert. De plus, l'exploration périodique des connaissances de l'ontologie permettent d'améliorer les performances des modèles de ML. Les performances de l'architecture ont été évaluées et validées sur plusieurs jeux de données UCI reliés à plusieurs problématiques industrielles de production de maintenance, ainsi que sur des bases de données issues des opérations de serrage réelles. Ceci a permis de démontrer la généralité et la robustesse de l'approche. Dans ce qui suit, la partie état de l'art et les contributions de cette thèse seront résumées, et les perspectives pour les travaux futurs seront présentées.

La partie état de l'art a permis d'identifier les problématiques associées à la surveillance process ainsi qu'à se positionner par rapport aux travaux existants. De ce fait, les méthodes de détection de défauts ont été présentées, en mettant en avant les méthodes basées sur les

---

données, du fait de leurs avantages. Puis, nous avons identifié les problématiques liées à ces méthodes et nous avons fait valoir que les modèles à mélanges de gaussiennes (GMM) peuvent être utilisées pour y remédier. Ensuite, la présentation détaillée des GMM avec un intérêt centré autour des mécanismes de décisions avec rejets et des méthodes de réduction de modèles (GMR) a été fournie. Les mécanismes de décisions avec rejets permettent de détecter de nouveaux événements et les méthodes de GMR permettent de réguler les GMM dans le temps, en termes de nombre de composantes. À l'issue de la présentation des GMM, nous avons discuté des limites que peuvent avoir ces algorithmes lorsqu'ils sont appliqués en industrie, et nous avons proposé une ontologie comme complément. À cet effet, la partie de l'état de l'art a été clôturée par la présentation des notions de base de l'ontologie et l'application de ces dernières dans le cadre du diagnostic de défauts (détection y compris). Nous avons montré que certaines tendances récentes consistent à mixer les techniques d'apprentissage automatique (ML) et de gestion des connaissances (KM) avec les ontologies pour une meilleure surveillance process. De ce fait, les travaux existants ainsi leurs limites ont été présentés et discutés. Ces limites ont servi de base pour nos contributions.

La deuxième partie du manuscrit décrit comment les problématiques énoncées dans la première partie ont été adressées. Nous avons commencé par présenter une nouvelle architecture hybride Ontologie-ML pour la découverte itérative de connaissances pour la détection de défauts en cours de fabrication. Cette dernière repose sur des GMM appris en non-supervisé sur des données d'apprentissage associées à un contexte industriel particulier, et un mécanisme permettant de découvrir des défauts potentiellement inconnus. Ensuite, les résultats d'apprentissage sont interprétés par un expert du domaine et liés sémantiquement à une ontologie. Cette liaison sémantique consiste à définir les instances des concepts de l'ontologie résumant les informations apprises par le ML et les connaissances de l'expert. Une fois l'ontologie initiée sur un contexte, la découverte itérative de connaissances associées aux nouveaux défauts détectés peut être enclenchée. De ce fait, l'architecture proposée élimine le besoin d'étiqueter des milliers de données pour apprendre un modèle de surveillance process. De plus, l'entraînement de ce dernier est soutenu par un expert du domaine qui permet de définir les caractéristiques ainsi que d'interpréter les résultats d'apprentissage, ce qui ajoute une confiance importante vis à vis des industriels. En outre, comme les GMM permettent d'approximer une distribution de données quelconque, cela donne à notre architecture un caractère générique. Par ailleurs, la capacité de l'architecture à détecter et intégrer de nouveaux défauts en cours de pro-

---

duction lui permet de suivre l'évolution du processus de fabrication au cours du temps. Enfin, l'architecture permet de gérer plusieurs contextes industriels simultanément, ce qui lui donne une capacité de gestion efficace, et de ce fait, elle peut être facilement implémentée en industrie. Les performances de l'architecture ont été prouvées et validées sur plusieurs jeux de données UCI (chapitre 4) ainsi que sur les données issues de serrages réelles (chapitre 6, section 6.2), montrant ainsi l'augmentation de la qualité du monitoring et l'intérêt de notre architecture.

Le second volet de nos contributions a consisté à fournir la capacité à notre architecture d'inspecter les connaissances ajoutées itérativement pour l'amélioration du système de détection de défauts. Pour ce faire, deux mécanismes de généralisation de connaissances ont été proposés. Le premier consiste à réduire le nombre de gaussiennes (dans un même contexte) afin de diminuer le sur-apprentissage ainsi que d'éliminer l'obsolescence et la redondance des gaussiennes. Le second consiste à généraliser les signatures de défaut issues de plusieurs contextes similaires. Cette généralisation permet au système de classer les observations dans un nouveau contexte industriel inconnu. Les deux mécanismes se basent sur la réduction de modèles (GMR) pour la généralisation de connaissances. L'algorithme de Réduction de Modèle avec Clustering (GMRC) utilisant la divergence de Kullback (KLD) a été choisi à cet effet. De plus, une heuristique a été proposée pour estimer le nombre de gaussiennes cible, qui est un élément clé pour l'utilisation de tout algorithme de GMR. Deux protocoles expérimentaux ont été proposés pour tester la performance de la généralisation de connaissances. Le premier consiste à valider la réduction intra-contextes suivant deux scénarios d'apprentissages itératifs réalistes. Les résultats ont montré que la réduction intra-contextes permet de réduire le sur-apprentissage, offrant ainsi une meilleure capacité de détection de défauts. Le second protocole consiste à valider la généralisation inter-contextes. Les premiers résultats ont montré que la généralisation inter-contextes améliore les performances de l'architecture (en termes de classification) en fonction du nombre de contextes connus, offrant ainsi le potentiel d'améliorer le système de détection en exploitant plusieurs contextes simultanément. Les seconds résultats ont montré que la généralisation permet d'améliorer la classification sur des contextes inconnus où peu de données existent, offrant ainsi la capacité à classer des observations issues de nouveaux contextes industriels inconnus. Toutes les expérimentations ont été réalisées sur des données UCI (chapitre 5) ainsi que sur des données issues d'opérations de serrage réelles (chapitre 6, section 6.3). Globalement, la généralisation des connaissances offre une propriété innovante à notre architecture qui consiste à améliorer les performances des

---

techniques de ML utilisées dans la détection de défauts dans les procédés de fabrication, en se basant sur les connaissances enregistrées dans l'ontologie.

Dans la dernière section de cette thèse, des perspectives sont proposées afin d'améliorer le travail actuel.

## Perspectives

Les travaux réalisés dans cette thèse ont permis de confirmer l'intérêt de combiner les techniques d'apprentissage automatique (ML) soutenues par un expert et la gestion des connaissances (KM) dans la surveillance de processus de fabrication. Toutefois, l'architecture proposée possède encore plusieurs axes d'amélioration conduisant à des perspectives. Pour commencer, une analyse approfondie sur les choix des techniques choisies sur différentes briques fonctionnelles de l'architecture pourrait être entreprise. En effet, l'objectif initial de cette thèse était d'implémenter l'architecture d'intelligence artificielle combinant ML et ontologie de bout en bout afin de la faire valider. Ce choix d'implémentation a conduit à des choix pratiques dans certaines briques de l'architecture comme : la sélection du nombre de cluster par le BIC [59], le critère de rejet sur une modification du critère d'Engel [53], et à une heuristique simple du critère de sélection de gaussiennes cible dans la réduction de modèles GMRC (section 5.2.4). À cet effet, il convient naturellement de comparer à court terme l'impact qu'auraient des choix alternatifs sur la performance globale de l'architecture. Cette comparaison pourrait apporter plus d'informations sur la sensibilité et le comportement de l'architecture suivant les différents choix, ce qui aide ainsi à guider les utilisateurs à mieux paramétrer les choix selon leurs préférences.

Une deuxième perspective est plus directement liée aux limites de la généralisation de connaissances. En effet, comme évoqué au chapitre 5, la généralisation de signatures peut affecter négativement les performances du système de détection de défauts dans le cas où les distributions associées à deux défauts différents se superposent. À cet effet, il serait intéressant de proposer un mécanisme de vérification permettant d'alerter l'utilisateur que deux classes de défaut semblent similaires et pourraient être fusionnées, sans quoi qu'une généralisation sur les signatures apprises pourrait détériorer le système de détection de défauts. Pour ce faire, les signatures de défauts issues de la généralisation pourraient être comparées. L'objectif est de détecter une superposition importante entre les signatures généralisées de défauts différents. Pour ce faire, une des mesures de dis-

---

similarité (mesures D) présentées à la section 2.3 pourrait être utilisée pour mesurer la divergence entre les signatures inter et intra-défauts (KLD par exemple). La divergence intra-défauts indiquerait à quel point les signatures généralisées d'un même défaut sont proches. La divergence inter-défauts indiquerait à quel point les signatures généralisées de différents défauts sont éloignées. Ainsi, si les mesures intra-défauts et l'inter-défauts étaient proches, nous aurions une forte indication que les signatures généralisées de plusieurs défauts soient superposées. Cette indication permettrait d'alerter sur la proximité de deux classes de défauts, et par conséquent, lui recommander trois actions. La première consiste à lui proposer une fusion des deux types de défauts détectés s'ils s'avèrent être de la même famille. La deuxième lui propose de rejeter cette généralisation, s'il tient à pouvoir distinguer ces 2 types de défauts. La dernière consiste à lui proposer de retravailler sur ses caractéristiques, pour mieux distinguer ces 2 défauts. Nous laissons ce travail en perspective. Enfin, il serait intéressant de comparer les performances de cette méthode suivant différentes mesures D.

Une troisième perspective consiste à étudier en profondeur la sensibilité de l'architecture quant au déséquilibre des classes de défauts lors de la généralisation des connaissances. En effet, nous avons constaté que l'architecture a été plus performante sur les jeux de données UCI (chapitre 5), mais moins performante sur les données de visage (chapitre 6). Ceci nous a conduit à augmenter les observations de la classe minoritaire. À cet effet, il serait intéressant d'étudier la sensibilité de l'architecture vis à vis du pourcentage de déséquilibre des classes. Une approche peut consister à tester suivant différents pourcentages de déséquilibre et constater à quel moment l'architecture perdrait en performance. Ensuite, si la sensibilité s'avère être un autre problème important, nous pouvons nous inspirer des travaux d'Adaptation au Domaine qui ont traité les déséquilibres des classes [166]. En effet, l'enjeu consistera donc à identifier des méthodes applicables aux GMM permettant de traiter les déséquilibres des classes par analogie aux méthodes citées [166].

Une quatrième perspective consiste à relaxer la contrainte formulée lors de la définition de la généralisation des connaissances. En effet, nous avons illustré au chapitre 5 que la généralisation multi-contextes exige que les signatures soient liées à la même classe de défaut sur tous les contextes. En effet, la généralisation ne pourra pas être enclenchée si les signatures sont associées à des classes de défauts différents. Cette contrainte est forte dans le sens où il faut qu'un défaut soit connu sur au moins deux contextes afin de pouvoir définir des signatures généralisées associées à ce défaut. Par ailleurs, nous avons démontré à travers les expériences que la disponibilité de plusieurs contextes ayant des signatures

---

d'un défaut identique permettait d'apprendre des signatures généralisées de meilleure qualité en termes de performance. Par conséquent, l'idéal serait que des gaussiennes associées à un même défaut soient apprises dans plusieurs contextes. Néanmoins, comme montré dans le chapitre 6, c'est difficilement le cas en industrie. À cet effet, il convient d'étudier la possibilité de générer des gaussiennes d'un défaut particulier à partir de la seule connaissance de la signature de l'état 'correct' dans un nouveau contexte, pour ainsi pouvoir transférer la connaissance des autres défauts connus. Cela permettra d'estimer la forme des signatures d'un défaut particulier sur tous les autres contextes à partir de la signature de la classe correcte dans chaque contexte ainsi que celle du défaut dans le contexte ayant le défaut. Cette situation est beaucoup plus réaliste industriellement car le taux d'opérations de fabrication correctes en industrie est globalement très élevé. En d'autres termes, il est plus facile de disposer de signatures associées à la classe correcte sur tous les contextes. Pour ce faire, nous pouvons de s'inspirer des travaux d'adaptation de domaines [165, 142] pour estimer les signatures d'un défaut dans chaque contexte. Ensuite, ces signatures pourront être exploitées par le mécanisme de généralisation pour définir des signatures généralisées pouvant être héritées par des contextes inconnus.

Enfin, une dernière perspective consiste à compléter l'architecture par l'exploitation de l'ontologie pour effectuer du diagnostic de défauts. En effet, rappelons que le raisonneur qui est un élément central de l'ontologie, et qui n'a pas été exploité dans l'architecture, ce qui n'enlève rien de son potentiel. Par conséquent, il convient naturellement de s'inspirer des travaux existants pour compléter l'architecture par le diagnostic de défauts et de ne pas se limiter à leurs détections seulement. À cet effet, l'enrichissement la structure de l'ontologie pourrait être effectué en ajoutant des concepts et sous concepts associés au diagnostic de défauts. Par exemple, en s'inspirant de Zhou et al [181], un concept Maintenance (ou Mesure de Correction) et un concept Cause de Défaut pourraient être proposés afin de définir des programmes de Maintenance (ou Mesure de Correction) pour défaut survenu par une cause particulière. Ces concepts serviront en outre à combler les parties manquantes du processus de diagnostic de défauts. En effet, le concept Cause de Défaut servira à définir précisément la source du défaut, ainsi qu'à identifier le degré de sa sévérité. Ensuite, le concept Maintenance (ou Mesure de Correction) proposera des programmes de Maintenance (ou Mesures de Correction) adaptées au défaut détecté. D'autre part, les signatures enregistrées pourraient être exploitées pour affiner les mesures de similarité entre les défauts. En effet, comme il existe un panel de mesures D pour comparer les signatures, ces dernières peuvent être utilisées pour mesurer la similarité

---

entre les défauts à partir de leurs signatures. Une forte similarité entre des signatures issues de défauts différents peut être un indicateur que ces défauts sont similaires ou proviennent de la même source. Cette information peut éclairer l'ontologie pour inférer la Cause et/ou le programme de Maintenance adéquat. Pour ce faire, la définition des règles SWRL combinant les critères de similarités entre les signatures et ceux provenant de l'ontologie pourrait être entamée. Cette idée d'exploiter l'ontologie avec le ML pour définir des règles SWRL afin effectuer le diagnostic de défauts rejoint en partie les travaux antérieurs, avec une touche supplémentaire de comparaison et généralisation de signatures qui ouvre un large potentiel pour l'amélioration du diagnostic de défauts dans les systèmes de fabrication.





# BIBLIOGRAPHIE

---

- [1] Anam ABID, Muhammad Tahir KHAN et Javaid IQBAL, « A review on fault detection and diagnosis techniques : basics and beyond », in : *Artificial Intelligence Review* 54.5 (2021), p. 3639-3664, ISSN : 15737462, DOI : 10.1007/s10462-020-09934-2, URL : <https://doi.org/10.1007/s10462-020-09934-2>.
- [2] Charu C. AGGARWAL, *Outlier analysis*, Springer, 2013, p. 1-446, ISBN : 9781461463962, DOI : 10.1007/978-1-4614-6396-2.
- [3] A. AKBARI, A. SETAYESHMEHR, H. BORSI, E. GOCKENBACH et I. FOFANA, « Intelligent agent-based system using dissolved gas analysis to detect incipient faults in power transformers », in : *IEEE Electrical Insulation Magazine* 26.6 (2010), p. 27-40, DOI : 10.1109/MEI.2010.5599977.
- [4] K ALTHOEFER, B LARA, Y H ZWEIRI et L D SENEVIRATNE, « Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks », in : *Proceedings of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science* 222.6 (2008), p. 1081-1095, DOI : 10.1243/09544062JMES546, eprint : <https://doi.org/10.1243/09544062JMES546>, URL : <https://doi.org/10.1243/09544062JMES546>.
- [5] Tohid ARDESHIRI, Umut ORGUNER et Emre ÖZKAN, « Gaussian Mixture Reduction Using Reverse Kullback-Leibler Divergence », in : (2015), p. 1-11, arXiv : 1508.05514, URL : <http://arxiv.org/abs/1508.05514>.
- [6] M. ASHBURNER, C.A. BALL, Judith BLAKE, David BOTSTEIN, Heather BUTLER et J. CHERRY, « Gene ontology : Tool for the unification of biology », in : *The Gene Ontology Consortium. Nat Genet* 25 (jan. 2000), p. 25-29.
- [7] Akbar ASSA et Konstantinos N. PLATANIOTIS, « Wasserstein-Distance-Based Gaussian Mixture Reduction », in : *IEEE Signal Processing Letters* 25.10 (2018), p. 1465-1469, DOI : 10.1109/LSP.2018.2865829.

- 
- [8] Mark A. AUSTIN, Parastoo DELGOSHAELI, Maria COELHO et Mohammad HEIDARINEJAD, « Architecting Smart City Digital Twins : Combined Semantic Model and Machine Learning Approach », in : *Journal of Management in Engineering* 36 (2020), p. 04020026.
- [9] Paul BEIER, Kenneth P. BURNHAM et David R. ANDERSON, *Model Selection and Inference : A Practical Information-Theoretic Approach*, t. 65, 3, 2001, p. 606, ISBN : 0387953647, DOI : 10.2307/3803117.
- [10] Kamel BENOETHMAN, Didier MAQUIN, José RAGOT et Mohamed BENREJEB, « Diagnosis of uncertain linear systems : an interval approach », in : *International Journal on Sciences and Techniques of Automatic control & computer engineering* 1.2 (déc. 2007), ISSN : 1737-7749, p. 136-154, URL : <https://hal.archives-ouvertes.fr/hal-00204963>.
- [11] Dimitri P. BERTSEKAS, *Nonlinear programming*, eng, Second edition, Belmont (Mass.) : Athena Scientific, 1999, ISBN : 1-886529-00-0.
- [12] C. BIERNACKI, G. CELEUX et G. GOVAERT, « Assessing a mixture model for clustering with the integrated completed likelihood », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.7 (2000), p. 719-725, ISSN : 1939-3539, DOI : 10.1109/34.865189, URL : <https://doi.org/10.1109/34.865189>.
- [13] Elnaz BIGDELI, Mahdi MOHAMMADI, Bijan RAAHEMI et Stan MATWIN, « Arbitrary shape cluster summarization with Gaussian mixture model », in : *KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (2014), p. 43-52, DOI : 10.5220/0005071500430052.
- [14] Samuel BLACKMAN et Robert POPOLI, *Design and Analysis of Modern Tracking*, Artech House, 1999.
- [15] K. BLEKAS, A. LIKAS, N. P. GALATSANOS et I. E. LAGARIS, « A spatially constrained mixture model for image segmentation », in : *IEEE Transactions on Neural Networks* 16.2 (2005), p. 494-498, ISSN : 1941-0093, DOI : 10.1109/TNN.2004.841773, URL : <https://doi.org/10.1109/TNN.2004.841773>.
- [16] Abdelhamid BOUCHACHIA et Charlie VANARET, « Incremental learning based on growing Gaussian mixture models », in : *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011* 2 (2011), p. 47-52, DOI : 10.1109/ICMLA.2011.79.

- 
- [17] P. BOURQUE, R. DUPUIS, A. ABRAN, J.W. MOORE et L. TRIPP, « The guide to the Software Engineering Body of Knowledge », in : *IEEE Software* 16.6 (1999), p. 35-44, DOI : 10.1109/52.805471.
- [18] Hamparsum BOZDOGAN, « Model selection and Akaike's Information Criterion (AIC) : The general theory and its analytical extensions », in : *Psychometrika* 52.3 (sept. 1987), p. 345-370, ISSN : 1860-0980, DOI : 10.1007/BF02294361, URL : <https://doi.org/10.1007/BF02294361>.
- [19] Kay Henning BRODERSEN, Cheng Soon ONG, Klaas Enno STEPHAN et Joachim M. BUHMANN, « The Balanced Accuracy and Its Posterior Distribution », in : *2010 20th International Conference on Pattern Recognition*, 2010, p. 3121-3124, DOI : 10.1109/ICPR.2010.764.
- [20] Baoping CAI, Hanlin LIU et Min XIE, « A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks », in : *Mechanical Systems and Signal Processing* 80 (2016), p. 31-44.
- [21] Qiushi CAO, Ahmed SAMET, Cecilia ZANNI-MERK, François De Bertrand DE BEUVRON et Christoph REICH, « An ontology-based approach for failure classification in predictive maintenance using fuzzy c-means and SWRL rules », in : *Procedia Computer Science* 159 (2019), p. 630-639, ISSN : 18770509, DOI : 10.1016/j.procs.2019.09.218.
- [22] Xiaopeng CAO, Jun LIU, Fanku MENG, Bo YAN, Hong ZHENG et Hongyi SU, « Anomaly detection for screw tightening timing data with LSTM recurrent neural network », in : *Proceedings - 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2019* (2019), p. 348-352, DOI : 10.1109/MSN48538.2019.00072.
- [23] Silvana CASTANO, Alfio FERRARA et Guillermo Nudelman HESS, « Discovery-Driven Ontology Evolution », in : *SWAP*, 2006.
- [24] Alfonso CASTRO, Andrés SEDANO, Javier GARCIA-ALGARRA, Eduardo VILLOSLADA et Victor VILLAGRA, « Application of a Multimedia Service and Resource Management Architecture for Fault Diagnosis », in : *Sensors* 18 (déc. 2017), p. 68, DOI : 10.3390/s18010068.

- 
- [25] Gilles CELEUX, Stephane CHRETIEN, Florence FORBES et Abdallah MKHADRI, « A Component-Wise EM Algorithm for Mixtures », in : *Journal of Computational and Graphical Statistics* 10.4 (2001), p. 697-712, DOI : 10.1198/106186001317243403, eprint : <https://doi.org/10.1198/106186001317243403>, URL : <https://doi.org/10.1198/106186001317243403>.
- [26] Gilles CELEUX et Gilda SOROMENHO, « An entropy criterion for assessing the number of clusters in a mixture model », in : *Journal of Classification* 13.2 (sept. 1996), p. 195-212, ISSN : 1432-1343, DOI : 10.1007/BF01246098, URL : <https://doi.org/10.1007/BF01246098>.
- [27] Faicel CHAMROUKHI, « Unsupervised learning of regression mixture models with unknown number of components », in : *Journal of Statistical Computation and Simulation* 86.12 (2016), p. 2308-2334, ISSN : 15635163, DOI : 10.1080/00949655.2015.1109096, arXiv : 1409.6981.
- [28] Faicel CHAMROUKHI et Bao-Tuyen HUYNH, « Regularized Maximum Likelihood Estimation and Feature Selection in Mixtures-of-Experts Models », in : *Journal de la Societe Française de Statistique* 160.1 (2019), URL : <https://hal.archives-ouvertes.fr/hal-02152434>.
- [29] Manuel Arias CHAO, Chetan KULKARNI, Kai GOEBEL et Olga FINK, « Hybrid deep fault detection and isolation : Combining deep neural networks and system performance models », in : *arXiv* (2019), p. 1-25, DOI : 10.48550/ARXIV.1908.01529, URL : <https://arxiv.org/abs/1908.01529>.
- [30] H. D. CHEN, K. C. CHANG et Chris SMITH, « Constraint optimized weight adaptation for Gaussian mixture reduction », in : *Signal Processing, Sensor Fusion, and Target Recognition XIX*, sous la dir. d'Ivan KADAR, t. 7697, International Society for Optics et Photonics, SPIE, 2010, p. 281-290, DOI : 10.1117/12.851993, URL : <https://doi.org/10.1117/12.851993>.
- [31] Rong CHEN, Zude ZHOU, Quan LIU, Duc Truong PHAM, Yuanyuan ZHAO, Junwei YAN et Qin WEI, « Knowledge modeling of fault diagnosis for rotating machinery based on ontology », in : *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, p. 1050-1055, DOI : 10.1109/INDIN.2015.7281880.

- 
- [32] Xianyi CHENG, Zhenzhong JIA, Ankit BHATIA, Reuben M. ARONSON et Matthew T. MASON, « Sensor Selection and Stage Result Classifications for Automated Miniature Screwdriving », in : *IEEE International Conference on Intelligent Robots and Systems* (2018), p. 6078-6085, ISSN : 21530866, DOI : 10.1109/IRoS.2018.8593520.
- [33] Sejune CHEON, Hankang LEE, Chang Ouk KIM et Seok Hyung LEE, « Convolutional Neural Network for Wafer Surface Defect Classification and the Detection of Unknown Defect Class », in : *IEEE Transactions on Semiconductor Manufacturing* 32.2 (2019), p. 163-170, ISSN : 15582345, DOI : 10.1109/TSM.2019.2902657.
- [34] David F. CROUSE, Peter WILLETT, Krishna PATTIPATI et Lennart SVENSSON, « A look at Gaussian mixture reduction algorithms », in : *Fusion 2011 - 14th International Conference on Information Fusion* (2011).
- [35] Shiyong CUI et Mihai DATCU, « Comparison of Kullback-Leibler divergence approximation methods between Gaussian mixture models for satellite image retrieval », in : *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (2015), p. 3719-3722.
- [36] *CWRU Bearing Dataset*, [https://www.kaggle.com/datasets/brjapon/cwru-bearing-datasets?resource=download&select=feature\\_time\\_48k\\_2048\\_load\\_1.csv](https://www.kaggle.com/datasets/brjapon/cwru-bearing-datasets?resource=download&select=feature_time_48k_2048_load_1.csv).
- [37] Alessandro D'ORTENZIO et Costanzo MANES, « Composite Transportation Dissimilarity in Consistent Gaussian Mixture Reduction », in : *Proceedings of 2021 IEEE 24th International Conference on Information Fusion, FUSION 2021 November* (2021), DOI : 10.23919/fusion49465.2021.9627011.
- [38] Alessandro D'ORTENZIO et Costanzo MANES, « Likeness-Based dissimilarity measures for Gaussian Mixture Reduction and Data Fusion », in : *Proceedings of 2021 IEEE 24th International Conference on Information Fusion, FUSION 2021 November* (2021).
- [39] C. DA CUNHA, B. AGARD et A. KUSIAK, « Data mining for improvement of product quality », in : *International Journal of Production Research* 44.18-19 (2006), p. 4027-4041, ISSN : 00207543, DOI : 10.1080/00207540600678904.

- 
- [40] C. DA CUNHA, B. AGARD et A. KUSIAK, « Selection of modules for mass customisation », in : *International Journal of Production Research* 48.5 (2010), p. 1439-1454, ISSN : 00207543, DOI : 10.1080/00207540802473989.
- [41] Julie DELON et Agnès DESOLNEUX, « A Wasserstein-type distance in the space of Gaussian Mixture Models », in : *SIAM Journal on Imaging Sciences* 13.2 (2020), p. 936-970, URL : <https://hal.archives-ouvertes.fr/hal-02178204>.
- [42] A. P. DEMPSTER, N. M. LAIRD et D. B. RUBIN, « Maximum likelihood from incomplete data via the EM algorithm », in : *Journal of the Royal Statistical Society : Series B* 39 (1977), p. 1-38.
- [43] Nadjette DENDANI-HADIBY et Mohamed Tarek KHADIR, « A Case based Reasoning System based on Domain Ontology for Fault Diagnosis of Steam Turbines », in : *International Journal of Hybrid Information Technology*, 2012.
- [44] Li DENG, Geoffrey HINTON et Brian KINGSBURY, « New types of deep neural network learning for speech recognition and related applications : an overview », in : *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, p. 8599-8603, DOI : 10.1109/ICASSP.2013.6639344.
- [45] Christian DETERS, Hak-Keung LAM, Emanuele Lindo SECCO, Helge A. WÜRDEMANN, Lakmal D. SENEVIRATNE et Kaspar ALTHOEFER, « Accurate Bolt Tightening Using Model-Free Fuzzy Control for Wind Turbine Hub Bearing Assembly », in : *IEEE Transactions on Control Systems Technology* 23.1 (2015), p. 1-12, DOI : 10.1109/TCST.2014.2309854.
- [46] P. DEVUVER et J. KITTLER, « A review of : “Pattern Recognition : A Statistical Approach” », in : *International Journal of Remote Sensing* 5.2 (1984), p. 464-464, DOI : 10.1080/01431168408948821, eprint : <https://doi.org/10.1080/01431168408948821>, URL : <https://doi.org/10.1080/01431168408948821>.
- [47] Javier DIAZ-ROZO, Concha BIELZA et Pedro LARRANAGA, « Clustering of Data Streams with Dynamic Gaussian Mixture Models : An IoT Application in Industrial Processes », in : *IEEE Internet of Things Journal* 5.5 (2018), p. 3533-3547, ISSN : 23274662, DOI : 10.1109/JIOT.2018.2840129.
- [48] Javier DIAZ-ROZO, Concha BIELZA et Pedro LARRAÑAGA, « Machine-tool condition monitoring with Gaussian mixture models-based dynamic probabilistic clustering », in : *Engineering Applications of Artificial Intelligence* 89.March 2019

- 
- (2020), p. 103434, ISSN : 09521976, DOI : 10.1016/j.engappai.2019.103434, URL : <https://doi.org/10.1016/j.engappai.2019.103434>.
- [49] Bertsimas DIMITRIS, *Introduction to linear optimization*, eng, Athena scientific series in optimization and neural computation, Belmont (Mass.) : Athena Scientific, 1997, ISBN : 1-886529-19-1.
- [50] Dheeru DUA et Casey GRAFF, *UCI Machine Learning Repository*, 2017, URL : <http://archive.ics.uci.edu/ml>.
- [51] Ricardo DUNIA, S. Joe QIN, Thomas F. EDGAR et Thomas J. McAVOY, « Identification of faulty sensors using principal component analysis », in : *Aiche Journal* 42 (1996), p. 2797-2812.
- [52] Shantanu DUTT, Federico ROTA, Franco TROVO et Fran HANCHEK, « Fault Tolerance in Computer Systems—From Circuits to Algorithms », in : *The Electrical Engineering Handbook*, sous la dir. de WAI-KAI CHEN, Burlington : Academic Press, 2005, p. 427-457, ISBN : 978-0-12-170960-0, DOI : <https://doi.org/10.1016/B978-012170960-0/50034-7>, URL : <https://www.sciencedirect.com/science/article/pii/B9780121709600500347>.
- [53] Paulo Martins ENGEL et Milton Roberto HEINEN, « Incremental learning of multivariate Gaussian mixture models », in : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6404 LNAI (2010), p. 82-91, ISSN : 03029743, DOI : 10.1007/978-3-642-16138-4\_9.
- [54] W.Y. ERAS-HERRERA, A.R. MESQUITA et B.O.S. TEIXEIRA, « Multiple-model multiple-hypothesis filter with Gaussian mixture reduction », in : *International Journal of Adaptive Control and Signal Processing* 32.2 (2018), p. 286-300, DOI : <https://doi.org/10.1002/acs.2841>, eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/acs.2841>, URL : <https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.2841>.
- [55] Ahmad FAOUR, Philippe LERAY et Bassam ETER, « Evolutivité d’une architecture en temps réel de filtrage d’alertes générées par les systèmes de détection d’intrusions sur les réseaux », in : *RFIA 2008*, Amiens, France, 2008, CDROM, URL : <https://hal.archives-ouvertes.fr/hal-00412885>.



- 
- [56] Usama FAYYAD, Gregory PIATETSKY-SHAPIO et Padhraic SMYTH, « From Data Mining to Knowledge Discovery in Databases », in : *AI Magazine* 17.3 (mars 1996), p. 37, DOI : 10.1609/aimag.v17i3.1230, URL : <https://ojs.aaai.org/index.php/aimagazine/article/view/1230>.
- [57] Mahmoud FERHAT, Mathieu RITOU, Philippe LERAY et Nicolas LE DU, « Incremental discovery of new defects : application to screwing process monitoring », in : *CIRP Annals* 70.1 (2021), p. 369-372, ISSN : 0007-8506, DOI : <https://doi.org/10.1016/j.cirp.2021.04.026>, URL : <https://www.sciencedirect.com/science/article/pii/S0007850621000500>.
- [58] Olga FINK, Qin WANG, Markus SVENSÉN, Pierre DERSIN, Wan Jui LEE et Melanie DUCOFFE, « Potential, challenges and future directions for deep learning in prognostics and health management applications », in : *Engineering Applications of Artificial Intelligence* 92.January (2020), p. 103678, ISSN : 09521976, DOI : 10.1016/j.engappai.2020.103678, arXiv : 2005.02144, URL : <https://doi.org/10.1016/j.engappai.2020.103678>.
- [59] C. FRALEY et A. E. RAFTERY, « How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis », in : *The Computer Journal* 41.8 (jan. 1998), p. 578-588.
- [60] Barbara FURLETTI et Franco TURINI, « Knowledge discovery in ontologies », in : *Intelligent Data Analysis* 16 (jan. 2012), p. 513-534, DOI : 10.3233/IDA-2012-0536.
- [61] Aldo GANGEMI, Domenico PISANELLI et Geri STEVE, « Ontology Integration : Experiences with Medical Terminologies », in : *Formal Ontology in Information Systems*, sous la dir. de Nicola GUARINO, Ios Press, 1998, p. 46-98.
- [62] Robert X. GAO, Lihui WANG, Moneer HELU et Roberto TETI, « Big data analytics for smart factories of the future », in : *CIRP Annals* 69.2 (2020), p. 668-692, ISSN : 0007-8506, DOI : <https://doi.org/10.1016/j.cirp.2020.05.002>, URL : <https://www.sciencedirect.com/science/article/pii/S0007850620301359>.
- [63] Janos GERTLER, *Fault detection and diagnosis in engineering systems*, New York : Marcel Dekker, 1998, p. 504, DOI : <https://doi.org/10.1201/9780203756126>.
- [64] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE, *Deep Learning*, <http://www.deeplearningbook.org>, MIT Press, 2016.

- 
- [65] Karl GRANSTRÖM et Umut ORGUNER, « On the reduction of Gaussian inverse Wishart mixtures », in : *15th International Conference on Information Fusion, FUSION 2012* (2012), p. 2162-2169.
- [66] Tamas GROF et Peter BAUER, « Voting-based Fault Detection for Aircraft Position Measurements with Dissimilar Observations », in : *IFAC-PapersOnLine 53.2* (2020), 21st IFAC World Congress, p. 14724-14729, ISSN : 2405-8963, DOI : <https://doi.org/10.1016/j.ifacol.2020.12.1841>, URL : <https://www.sciencedirect.com/science/article/pii/S2405896320324551>.
- [67] Thomas R. GRUBER, « A translation approach to portable ontology specifications », in : *Knowledge Acquisition 5.2* (1993), p. 199-220, ISSN : 10428143, DOI : 10.1006/knac.1993.1008.
- [68] Tobias GRÜNER, Falco BÖLLHOFF, Robert MEISETZSCHLÄGER, Alexander VYDRENKO, Martyna BATOR, Alexander DICKS et Andreas THESSLER, « Evaluation of machine learning for sensorless detection and classification of faults in electromechanical drive systems », in : *Procedia Computer Science 176* (2020), p. 1586-1595.
- [69] Peter HAASE et Ljiljana" STOJANOVIC, « Consistent Evolution of OWL Ontologies" », in : *The Semantic Web : Research and Applications*", sous la dir. d'Asunción GÓMEZ-PÉREZ et Jérôme" EUZENAT, "Berlin, Heidelberg" : "Springer Berlin Heidelberg", 2005, "182-197", ISBN : "978-3-540-31547-6".
- [70] J. A. HARDING, M. SHAHBAZ, SRINIVAS et A. KUSIAK, « Data mining in manufacturing : A review », in : *Journal of Manufacturing Science and Engineering 128.4* (2006), p. 969-976, ISSN : 10871357, DOI : 10.1115/1.2194554.
- [71] Jeff HEFLIN, James HENDLER et Sean LUKE, « Coping with Changing Ontologies in a Distributed Environment », in : *Ontology Management. Papers from the AAAI Workshop. WS-99-13. AAAI Press* (juin 1999), p. 74-79.
- [72] Milton HEINEN, « A Connectionist Approach for Incremental Function Approximation and On-line Tasks », thèse de doct., Universidade Federal do Rio Grande do Sul, jan. 2011.
- [73] N. HELWIG, E. PIGNANELLI et A. SCHÜTZE, « D8.1 - Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System », in : *Proceedings SENSOR 2015 D8 - Signals & Systems* (2020), p. 641-646, DOI : 10.5162/sensor2015/D8.1.

- 
- [74] Kilian HENDRICKX, Lorenzo PERINI, Dries Van der PLAS, Wannes MEERT et Jesse DAVIS, « Machine Learning with a Reject Option : A survey », in : *ArXiv abs/2107.11277* (2021), DOI : 10.48550/ARXIV.2107.11277, URL : <https://arxiv.org/abs/2107.11277>.
- [75] Marco F. HUBER et Uwe D. HANEBECK, « Progressive Gaussian mixture reduction », in : *Proceedings of the 11th International Conference on Information Fusion, FUSION 2008* (2008).
- [76] Bao Tuyen HUYNH et Faicel CHAMROUKHI, « Estimation and Feature Selection in Mixtures of Generalized Linear Experts Models », in : *arXiv* (2019), DOI : 10.48550/ARXIV.1907.06994, URL : <https://arxiv.org/abs/1907.06994>.
- [77] Rolf ISERMANN, *Fault-Diagnosis Applications : Model-Based Condition Monitoring Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems*, 1st, Springer Publishing Company, Incorporated, 2011, ISBN : 3642127665.
- [78] « Iterative knowledge discovery for fault detection in manufacturing systems », in : *Knowledge-Based and Intelligent Information & Engineering Systems 26th Annual Conference, KES-2022* (2022), p. 1-10, URL : <https://hal-mines-paristech.archives-ouvertes.fr/UNAM/hal-03764231v1>.
- [79] Zhenzhong JIA, Ankit BHATIA, Reuben M. ARONSON, David BOURNE et Matthew T. MASON, « A Survey of Automated Threaded Fastening », in : *IEEE Transactions on Automation Science and Engineering* 16.1 (2019), p. 298-310, ISSN : 15455955, DOI : 10.1109/TASE.2018.2835382.
- [80] C. JORGE CHAMBY-DIAZ, Mariana RECAMONDE-MENDOZA, L. C. ANA BAZZAN et Ricardo GRUNITZKI, « Adaptive Incremental Gaussian Mixture Network for Non-Stationary Data Stream Classification », in : *Proceedings of the International Joint Conference on Neural Networks 2018-July.i* (2018), p. 1-8, DOI : 10.1109/IJCNN.2018.8489049.
- [81] Kittipat KAMPA, Erion HASANBELLIU et Jose PRINCIPE, « Closed-form cauchy-schwarz PDF divergence for mixture of Gaussians », in : *Proceedings of the International Joint Conference on Neural Networks* (juill. 2011), p. 2578-2585, DOI : 10.1109/IJCNN.2011.6033555.

- 
- [82] Samir KHAN et Takehisa YAIRI, « A review on the application of deep learning in system health management », in : *Mechanical Systems and Signal Processing* 107 (2018), p. 241-265, ISSN : 10961216, DOI : 10.1016/j.ymssp.2017.11.024, URL : <https://doi.org/10.1016/j.ymssp.2017.11.024>.
- [83] Dong-Hyeon KIM et al., « Smart Machining Process Using Machine Learning : A Review and Perspective on Machining Industry », in : *International Journal of Precision Engineering and Manufacturing-Green Technology* 5.4 (août 2018), p. 555-568, ISSN : 2198-0810, DOI : 10.1007/s40684-018-0057-y, URL : <https://doi.org/10.1007/s40684-018-0057-y>.
- [84] Michel KINNAERT, « Fault diagnosis based on analytical models for linear and nonlinear systems - a tutorial », in : *IFAC Proceedings Volumes* 36.5 (2003), 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes 2003, Washington DC, 9-11 June 1997, p. 37-50, ISSN : 1474-6670, DOI : [https://doi.org/10.1016/S1474-6670\(17\)36468-6](https://doi.org/10.1016/S1474-6670(17)36468-6), URL : <https://www.sciencedirect.com/science/article/pii/S1474667017364686>.
- [85] Kevin KNIGHT et Steve K. LUK, « Building a Large-Scale Knowledge Base for Machine Translation », in : *AAAI*, 1994.
- [86] Murali KRISHNAMURTHI et Don T. PHILLIPS, « An expert system framework for machine fault diagnosis », in : *Computers & Industrial Engineering* 22.1 (1992), p. 67-84, ISSN : 0360-8352, DOI : [https://doi.org/10.1016/0360-8352\(92\)90034-H](https://doi.org/10.1016/0360-8352(92)90034-H), URL : <https://www.sciencedirect.com/science/article/pii/036083529290034H>.
- [87] Li-Lin KU et Tung-Chen HUANG, « Sequential monitoring of manufacturing processes : an application of grey forecasting models », in : *The International Journal of Advanced Manufacturing Technology* 27.5 (jan. 2006), p. 543-546, ISSN : 1433-3015, DOI : 10.1007/s00170-004-2198-0, URL : <https://doi.org/10.1007/s00170-004-2198-0>.
- [88] S. KULLBACK et R. A. LEIBLER, « On Information and Sufficiency », in : *The Annals of Mathematical Statistics* 22.1 (1951), p. 79-86, DOI : 10.1214/aoms/1177729694, URL : <https://doi.org/10.1214/aoms/1177729694>.

- 
- [89] Asma LADJ, Zhiqiang WANG, Oussama MESKI, Farouk BELKADI, Mathieu RITOU et Catherine DA CUNHA, « A knowledge-based Digital Shadow for machining industry in a Digital Twin perspective », in : *Journal of Manufacturing Systems* 58 (2021), p. 168-179, ISSN : 02786125, DOI : 10.1016/j.jmsy.2020.07.018.
- [90] Jean Baptiste LAMY, « Owlready : Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies », in : *Artificial Intelligence in Medicine* 80 (2017), p. 11-28, ISSN : 18732860.
- [91] Jay LEE, Behrad BAGHERI et Hung-An KAO, « A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems », in : *Manufacturing Letters* 3 (2015), p. 18-23, ISSN : 2213-8463, DOI : <https://doi.org/10.1016/j.mfglet.2014.12.001>, URL : <https://www.sciencedirect.com/science/article/pii/S221384631400025X>.
- [92] Fritz LÉHMANN, « Machine-negotiated, ontology-based EDI (Electronic Data Interchange) », in : *Electronic Commerce*, sous la dir. de Nabil R. ADAM et Yelena YESHA, Berlin, Heidelberg : Springer Berlin Heidelberg, 1996, p. 27-45, ISBN : 978-3-540-49355-6.
- [93] Richard D. Braatz LEO H. CHIANG Evan L. Russell, *Fault detection and diagnosis in industrial systems*, 1994, p. 293, ISBN : 9781852333270, DOI : 10.1088/0957-0233/12/10/706, URL : <https://link.springer.com/book/10.1007/978-1-4471-0347-9>.
- [94] P. LERAY, H. ZARAGOZA et F. D'ALCHÉ-BUC, « Pertinence des Mesures de Confiance en Classification », in : *12ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA 2000)*, Paris, France, 2000, p. 267-276.
- [95] Linlin LI, *Fault Detection and Fault-Tolerant Control for Nonlinear Systems*, 2016, ISBN : 9783658130190, DOI : 10.1007/978-3-658-13020-6.
- [96] Wei LI, Minjun PENG et Qingzhong WANG, « Fault identification in PCA method during sensor condition monitoring in a nuclear power plant », in : *Annals of Nuclear Energy* 121 (2018), p. 135-145, ISSN : 0306-4549, URL : <https://www.sciencedirect.com/science/article/pii/S0306454918303761>.

- 
- [97] Weigui LI, Zhuqing YUAN, Wenyu SUN et Yongpan LIU, « Domain Adaptation for Intelligent Fault Diagnosis under Different Working Conditions », in : *MATEC Web of Conferences* 319 (2020), p. 03001, DOI : 10.1051/mateconf/202031903001.
- [98] Arunas LIPNICKAS, « Two-Stage Neural Networks Based Classifier System for Fault Diagnosis », in : *Computational Intelligence in Fault Diagnosis* (déc. 2006), p. 209-230, DOI : 10.1007/978-1-84628-631-5\_7.
- [99] Bin LIU, Zheyuan DING, Junfeng WU et Li YAO, « Ontology-based fault diagnosis : A decade in review », in : *ACM International Conference Proceeding Series* (2019), p. 112-116, DOI : 10.1145/3307363.3307381.
- [100] S. LLOYD, « Least squares quantization in PCM », in : *IEEE Transactions on Information Theory* 28.2 (1982), p. 129-137, DOI : 10.1109/TIT.1982.1056489.
- [101] Nick LORD, « Matrix computations, 3rd edition, by G. H. Golub and C. F. Van Loan. Pp. 694. 1996. £25 (paper), £54 (hard). ISBN 0 8018 5414 8, 0 8018 5413 X. (Johns Hopkins University Press). », in : *The Mathematical Gazette* 83.498 (1999), p. 556-557, DOI : 10.2307/3621013.
- [102] Clement LORK, Vishal CHOUDHARY, Naveed UL HASSAN, Wayes TUSHAR, Chau YUEN, Benny Kai Kiat NG, Xinyu WANG et Xiang LIU, « An Ontology-Based Framework for Building Energy Management with IoT », in : *Electronics (Switzerland)* 8.5 (2019), p. 1-15, ISSN : 20799292, DOI : 10.3390/electronics8050485.
- [103] Aleix M. MARTÍNEZ et Jordi VITRIÀ, « Learning mixture models using a genetic version of the EM algorithm », in : *Pattern Recognition Letters* 21.8 (2000), p. 759-769, ISSN : 0167-8655, DOI : [https://doi.org/10.1016/S0167-8655\(00\)00031-3](https://doi.org/10.1016/S0167-8655(00)00031-3), URL : <https://www.sciencedirect.com/science/article/pii/S0167865500000313>.
- [104] Takayuki MATSUNO, Jian HUANG et Toshio FUKUDA, « Fault detection algorithm for external thread fastening by robotic manipulator using linear support vector machine classifier », in : *IEEE International Conference on Robotics and Automation*, 2013, p. 3443-3450, DOI : 10.1109/ICRA.2013.6631058.
- [105] G. J. McLACHLAN et D. PEEL, *Finite mixture models*, New York : Wiley Series in Probability et Statistics, 2000.

- 
- [106] Gabriela MEDINA-OLIVA, Benoît IUNG, Luis BARBERÁ, Pablo VIVEROS et Thomas RUIN, « Root cause analysis to identify physical causes », in : *11th International Probabilistic Safety Assessment and Management Conference and The Annual European Safety and Reliability Conference, PSAM11 - ESREL 2012*, Helsinki, Finland, juin 2012, CDROM, URL : <https://hal.archives-ouvertes.fr/hal-00748696>.
- [107] Gabriela MEDINA-OLIVA, Alexandre VOISIN, Maxime MONNIN et Jean-Baptiste LÉGER, « Predictive Diagnosis based on a Fleet-wide Ontology Approach », in : *Knowledge-Based Systems* 68 (jan. 2013), DOI : 10.1016/j.knosys.2013.12.020.
- [108] Gabriel MICHAU et Olga FINK, « Unsupervised Fault Detection in Varying Operating Conditions », in : *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2019, p. 1-10, DOI : 10.1109/ICPHM.2019.8819383.
- [109] Gabriel MICHAU, Thomas PALME et Olga FINK, « Deep feature learning network for fault detection and isolation », in : *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM* (2017), p. 108-118, ISSN : 23250178.
- [110] Dubravko MILJKOVIC, « Fault Detection Using Limit Checking : A Brief Introductory Review », in : *44th International Convention on Information, Communication and Electronic Technology, MIPRO 2021, Opatija, Croatia, September 27 - Oct. 1, 2021*, IEEE, 2021, p. 908-913, DOI : 10.23919/MIPRO52101.2021.9596786, URL : <https://doi.org/10.23919/MIPRO52101.2021.9596786>.
- [111] Dubravko MILJKOVIĆ, « Fault detection methods : A literature survey », in : *MIPRO 2011 - 34th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings* (2011), p. 750-755.
- [112] Jacky MONTMAIN et Lydie LEYVAL, « Causal Graphs for Model Based Diagnosis », in : *IFAC Proceedings Volumes* 27.5 (1994), p. 329-337, ISSN : 14746670, DOI : 10.1016/S1474-6670(17)48049-9.
- [113] Guilherme R. MOREIRA, Gustavo J.G. LAHR, Thiago BOAVENTURA, Jose O. SAVAZZI et Glauco A.P. CAURIN, « Online prediction of threading task failure using Convolutional Neural Networks », in : *IEEE International Conference on Intelligent Robots and Systems* (2018), p. 2056-2061, ISSN : 21530866, DOI : 10.1109/IRoS.2018.8594501.

- 
- [114] William NELSON et Charles CULP, « Machine Learning Methods for Automated Fault Detection and Diagnostics in Building Systems—A Review », in : *Energies* 15.15 (déc. 2022), p. 1-20, URL : <https://ideas.repec.org/a/gam/jeners/v15y2022i15p5534-d876157.html>.
- [115] Natasha NOY et Michel KLEIN, « Ontology Evolution : Not the Same as Schema Evolution », in : *Knowledge and Information Systems* 6 (mars 2004), DOI : 10.1007/s10115-003-0137-2.
- [116] Marek OBITKO et Václav JIRKOVSKÝ, « Big Data Semantics in Industry 4.0 », in : *Industrial Applications of Holonic and Multi-Agent Systems*, sous la dir. de Vladimír MAŘÍK, Arnd SCHIRRMANN, Damien TRENTESAUX et Pavel VRBA, Cham : Springer International Publishing, 2015, p. 217-229.
- [117] Gustavo H.F.M. OLIVEIRA, Leandro L. MINKU et Adriano L.I. OLIVEIRA, « GMM-VRD : A Gaussian Mixture Model for Dealing with Virtual and Real Concept Drifts », in : *Proceedings of the International Joint Conference on Neural Networks* 2019-July (2019), DOI : 10.1109/IJCNN.2019.8852097.
- [118] Luan Soares OLIVEIRA et Gustavo E.A.P.A. BATISTA, « IGMM-CD : A Gaussian mixture classification algorithm for data streams with concept drifts », in : *Proceedings - 2015 Brazilian Conference on Intelligent Systems, BRACIS 2015 November* (2016), p. 55-61, DOI : 10.1109/BRACIS.2015.61.
- [119] Jonathan J. OLIVER, Rohan A. BAXTER et Chris S. WALLACE, « Unsupervised Learning Using MML », in : *ICML*, 1996.
- [120] Emeric OSTERMEYER, Christophe DANJOU, Alexandre DURUPT et Julien Le DUIGOU, « An ontology-based framework for the management of machining information in a data mining perspective », in : *IFAC-PapersOnLine* 51.11 (2018), p. 302-307, ISSN : 24058963, DOI : 10.1016/j.ifacol.2018.08.300, URL : <https://doi.org/10.1016/j.ifacol.2018.08.300>.
- [121] Lucy Y. PAO, « Multisensor multitarget mixture reduction algorithms for tracking », in : *Journal of Guidance Control and Dynamics* 17 (1994), p. 1205-1211.
- [122] R.J. PATTON et J. CHEN, « A Review of Parity Space Approaches to Fault Diagnosis », in : *IFAC Proceedings Volumes* 24.6 (1991), IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'91), Baden-Baden, Germany, 10-13 September 1991, p. 65-81, ISSN : 1474-



- 
- 6670, DOI : [https://doi.org/10.1016/S1474-6670\(17\)51124-6](https://doi.org/10.1016/S1474-6670(17)51124-6), URL : <https://www.sciencedirect.com/science/article/pii/S1474667017511246>.
- [123] Haim PERMUTER, Joseph FRANCOIS et Ian JERMYN, « A study of Gaussian mixture models of color and texture features for image classification and segmentation », in : *Pattern Recognition* 39 (avr. 2006), p. 695-706, DOI : 10.1016/j.patcog.2005.10.028.
- [124] Rafael PINTO et Paulo ENGEL, « Scalable and Incremental Learning of Gaussian Mixture Models », in : (2017), arXiv : 1701.03940, URL : <http://arxiv.org/abs/1701.03940>.
- [125] Rafael Coimbra PINTO et Paulo Martins ENGEL, « A fast incremental Gaussian mixture model », in : *PLoS ONE* 10.10 (2015), p. 1-12, ISSN : 19326203, DOI : 10.1371/journal.pone.0139931, arXiv : 1506.04422.
- [126] Peter PLESSERS et Olga DE TROYER, « Ontology Change Detection Using a Version Log », in : *The Semantic Web – ISWC 2005*, sous la dir. d'Yolanda GIL, Enrico MOTTA, V. Richard BENJAMINS et Mark A. MUSEN, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 578-592, ISBN : 978-3-540-32082-1.
- [127] Supawan PONPITAKCHAI, « Monitoring Screw Fastening Process : an Application of SVM Classification », in : *Naresuan University Engineering Journal* 11.1 (2016), p. 1-6, ISSN : 2651-1568, URL : <https://www.tci-thaijo.org/index.php/nuej/article/view/35154>.
- [128] David POOLE, Computer SCIENCE et British COLUMBIA, « Normality and Faults in Logic-Based Diagnosis », in : *ICML*, 1989, p. 1304-1310.
- [129] Jian QIN, Ying LIU et Roger GROSVENOR, « A Categorical Framework of Manufacturing for Industry 4.0 and Beyond », in : *Procedia CIRP* 52 (2016), The Sixth International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2016), p. 173-178, ISSN : 2212-8271, DOI : <https://doi.org/10.1016/j.procir.2016.08.005>, URL : <https://www.sciencedirect.com/science/article/pii/S221282711630854X>.
- [130] Dnyanesh RAJPATHAK, « An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain », in : *Computers in Industry* 64 (juin 2013), p. 565-580, DOI : 10.1016/j.compind.2013.03.001.

- 
- [131] Dnyanesh RAJPATHAK et Rahul CHOUGULE, « A generic ontology development framework for data integration and decision support in a distributed environment », in : *Robotics and Computer-Integrated Manufacturing* 24 (fév. 2011), p. 154-170, DOI : 10.1080/0951192X.2010.531291.
- [132] Dnyanesh RAJPATHAK, Halasya Siva SUBRAMANIA et Pulak BANDYOPADHYAY, « Ontology-driven data collection and validation framework for the diagnosis of vehicle health management », in : *International Journal of Computer Integrated Manufacturing* 25 (sept. 2012), p. 1-16, DOI : 10.1080/0951192X.2012.665187.
- [133] Philippe RAUFFET, Catherine DA CUNHA et Alain BERNARD, « Designing and managing organizational interoperability with organizational capabilities and roadmaps », in : *Proceedings - 2009 International Conference on Interoperability for Enterprise Software and Applications, IESA 2009* (2009), p. 120-126, DOI : 10.1109/I-ESA.2009.24.
- [134] *Real datasets with concept drift*, URL : <https://en.wikipedia.org/wiki/Concept%20drift#Real..>
- [135] D. A. REYNOLDS et R. C. ROSE, « Robust text-independent speaker identification using Gaussian mixture speaker models », in : *IEEE Transactions on Speech and Audio Processing* 3.1 (1995), p. 72-83, ISSN : 1558-2353, DOI : 10.1109/89.365379, URL : <https://doi.org/10.1109/89.365379>.
- [136] Douglas A. REYNOLDS, Thomas F. QUATIERI et Robert B. DUNN, « Speaker Verification Using Adapted Gaussian Mixture Models », in : *Digital Signal Processing* 10.1 (jan. 2000), p. 19-41, ISSN : 1051-2004, URL : <https://www.sciencedirect.com/science/article/pii/S1051200499903615>.
- [137] Diogo RIBEIRO, Luís Miguel MATOS, Guilherme MOREIRA, André PILASTRI et Paulo CORTEZ, « Isolation Forests and Deep Autoencoders for Industrial Screw Tightening Anomaly Detection », in : *Computers* 11.4 (2022), ISSN : 2073-431X, DOI : 10.3390/computers11040054, URL : <https://www.mdpi.com/2073-431X/11/4/54>.
- [138] Sylvia RICHARDSON et Peter GREEN, « On Bayesian analysis of mixture with unknown number of components », in : *J. R. Stat. Soc. Ser. B* 60 (nov. 1996), DOI : 10.2307/2985194.

- 
- [139] Jorma RISSANEN, « A Universal Prior for Integers and Estimation by Minimum Description Length », in : *The Annals of Statistics* 11.2 (1983), p. 416-431, DOI : 10.1214/aos/1176346150, URL : <https://doi.org/10.1214/aos/1176346150>.
- [140] Mathieu RITOU, Farouk BELKADI, Zakaria YAHOUNI, Catherine DA CUNHA, Florent LAROCHE et Benoit FURET, « Knowledge-based multi-level aggregation for decision aid in the machining industry », in : *CIRP Annals* 68.1 (2019), p. 475-478, ISSN : 0007-8506, DOI : <https://doi.org/10.1016/j.cirp.2019.03.009>, URL : <https://www.sciencedirect.com/science/article/pii/S0007850619300101>.
- [141] Kathryn ROEDER et Larry WASSERMAN, « Practical Bayesian Density Estimation Using Mixtures of Normals », in : *Journal of the American Statistical Association* 92.439 (sept. 1997), p. 894-902, ISSN : 0162-1459, DOI : 10.1080/01621459.1997.10474044, URL : <https://doi.org/10.1080/01621459.1997.10474044>.
- [142] Katharina ROMBACH, Gabriel MICHAU et Olga FINK, « Contrastive Learning for Fault Detection and Diagnostics in the Context of Changing Operating Conditions and Novel Fault Types », in : *Sensors* 21.10 (2021), ISSN : 1424-8220, DOI : 10.3390/s21103550, URL : <https://www.mdpi.com/1424-8220/21/10/3550>.
- [143] Andrew R. RUNNALLS, « Kullback-Leibler approach to Gaussian mixture reduction », in : *IEEE Transactions on Aerospace and Electronic Systems* 43.3 (2007), p. 989-999, ISSN : 00189251, DOI : 10.1109/TAES.2007.4383588.
- [144] D. J. SALMOND, « Mixture Reduction Algorithms for Point and Extended Object Tracking in Clutter », in : *IEEE Transactions on Aerospace and Electronic Systems* 45.2 (2009), p. 667-686, DOI : 10.1109/TAES.2009.5089549.
- [145] F. SAMIRMI, W.H. TANG et Q. WU, « Fuzzy Ontology Reasoning for Power Transformer Fault Diagnosis », in : *Advances in Electrical and Computer Engineering* 15 (nov. 2015), p. 107-114, DOI : 10.4316/AECE.2015.04015.
- [146] Dennis SCHIEFERDECKER et Marco F. HUBER, « Gaussian mixture reduction via clustering », in : *2009 12th International Conference on Information Fusion, FUSION 2009* (2009), p. 1536-1543.
- [147] David W SCOTT, « Parametric Statistical Modeling by Minimum Integrated Square Error », in : *Technometrics* 43.3 (2001), p. 274-285, DOI : 10.1198/004017001316975880, eprint : <https://doi.org/10.1198/004017001316975880>, URL : <https://doi.org/10.1198/004017001316975880>.

- 
- [148] Noman SHABBIR, Lauri KÜTT, Bilal ASAD, Muhammad JAWAD, Muhammad Naveed IQBAL et Kamran DANIEL, « Spectrum analysis for condition monitoring and fault diagnosis of ventilation motor : A case study », in : *Energies* 14.7 (2021), p. 1-16, ISSN : 19961073, DOI : 10.3390/en14072001.
- [149] Ralph S SHOBERG, « Engineering Fundamentals of Threaded Fastener Design and Analysis », in : *RS Technologies* (2000), p. 1-39, URL : <http://www.hexagon.de/rs/engineering%20fundamentals.pdf>.
- [150] *Smart Factory Machine Learning for Predictive Maintenance Testbed, Ind. Internet Consortium, Needham, MA, USA, 2017*, URL : <https://hub.iiconsortium.org/smart-factory-machine-learning>.
- [151] Wade A. SMITH et Robert B. RANDALL, « Rolling element bearing diagnostics using the Case Western Reserve University data : A benchmark study », in : *Mechanical Systems and Signal Processing* 64-65 (2015), p. 100-131, ISSN : 0888-3270, DOI : <https://doi.org/10.1016/j.ymssp.2015.04.021>, URL : <https://www.sciencedirect.com/science/article/pii/S0888327015002034>.
- [152] Padhraic SMYTH, « Model selection for probabilistic clustering using cross-validated likelihood », in : *Statistics and Computing* 10.1 (jan. 2000), p. 63-72, ISSN : 1573-1375, DOI : 10.1023/A:1008940618127, URL : <https://doi.org/10.1023/A:1008940618127>.
- [153] Hichem SNOUSSI, « Bayesian Information Geometry : Application to Prior Selection on Statistical Manifolds », in : sous la dir. de Peter HAWKES, t. 146, *Advances in Imaging and Electron Physics*, Elsevier, 2007, p. 163-207, DOI : [https://doi.org/10.1016/S1076-5670\(06\)46003-1](https://doi.org/10.1016/S1076-5670(06)46003-1), URL : <https://www.sciencedirect.com/science/article/pii/S1076567006460031>.
- [154] Ashish Kumar SRIVASTAVA, « Comparison Analysis of Machine Learning algorithms for Steel Plate Fault Detection », in : *International Research Journal of Engineering and Technology* 6.5 (2019), p. 1231-1234, ISSN : 2395-0072, URL : [www.irjet.net](http://www.irjet.net).
- [155] C. STAUFFER et W. E. L. GRIMSON, « Adaptive background mixture models for real-time tracking », in : *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, t. 2, Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Re-

- 
- cognition (Cat. No PR00149), 1999, 246-252 Vol. 2, DOI : 10.1109/CVPR.1999.784637, URL : <https://doi.org/10.1109/CVPR.1999.784637>.
- [156] Ljiljana STOJANOVIC, Alexander MAEDCHE, Nenad STOJANOVIC et Rudi STUDER, « Ontology evolution as reconfiguration-design problem solving », in : *K-CAP '03*, jan. 2003, p. 162, DOI : 10.1145/945649.945669.
- [157] Bowen SUN, Jiongqi WANG, Zhangming HE, Haiyin ZHOU et Fengshou GU, « Fault Identification for a Closed-Loop Control System Based on an Improved Deep Neural Network », in : *Sensors* 19 (mai 2019), p. 2131, DOI : 10.3390/s19092131.
- [158] Yulia SVETASHOVA, Baifan ZHOU, Tim PYCHYNSKI, Stefan SCHMIDT, York SUREVETTER, Ralf MIKUT et Evgeny KHARLAMOV, « Ontology-Enhanced Machine Learning : A Bosch Use Case of Welding Quality Monitoring », in : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12507 LNCS (2020), p. 531-550, ISSN : 16113349, DOI : 10.1007/978-3-030-62466-8\_33.
- [159] Dhaval THAKKER, Pankesh PATEL, Muhammad INTIZAR ALI, Tejal SHAH, Qiu-shi CAO, Ahmed SAMET, Cecilia ZANNI-MERK, François DE BERTRAND DE BEUVRON et Christoph REICH, « Combining chronicle mining and semantics for predictive maintenance in manufacturing processes », in : *Semantic Web* 11.6 (2020), p. 927-948, ISSN : 22104968, DOI : 10.3233/SW-200406.
- [160] Naonori UEDA et Ryohei NAKANO, « Deterministic annealing EM algorithm », in : *Neural Networks* 11.2 (1998), p. 271-282, ISSN : 0893-6080, DOI : [https://doi.org/10.1016/S0893-6080\(97\)00133-0](https://doi.org/10.1016/S0893-6080(97)00133-0), URL : <https://www.sciencedirect.com/science/article/pii/S0893608097001330>.
- [161] Naonori UEDA, Ryohei NAKANO, Zoubin GHAHRAMANI et Geoffrey E HINTON, « SMEM Algorithm for Mixture Models », in : *Advances in Neural Information Processing Systems*, sous la dir. de M. KEARNS, S. SOLLA et D. COHN, t. 11, MIT Press, 1998, URL : <https://proceedings.neurips.cc/paper/1998/file/253f7b5d921338af34da817c00f42753-Paper.pdf>.
- [162] Athanasios VOULODIMOS, Nikolaos DOULAMIS, Anastasios DOULAMIS et Eftychios PROTOPAPADAKIS, « Deep Learning for Computer Vision : A Brief Review », in : *Computational Intelligence and Neuroscience* 2018 (fév. 2018), p. 7068349, ISSN : 1687-5265, DOI : 10.1155/2018/7068349, URL : <https://doi.org/10.1155/2018/7068349>.

- 
- [163] Yuchai WAN, Xiabi LIU, Yi WU, Lunhao GUO, Qiming CHEN et Murong WANG, « ICGT : A novel incremental clustering approach based on GMM tree », in : *Data and Knowledge Engineering* 117 (2018), p. 71-86, ISSN : 0169023X, DOI : 10.1016/j.datak.2018.07.003.
- [164] Qin WANG, Gabriel MICHAU et Olga FINK, « Domain Adaptive Transfer Learning for Fault Diagnosis », in : *Proceedings - 2019 Prognostics and System Health Management Conference, PHM-Paris 2019* (2019), p. 279-285, DOI : 10.1109/PHM-Paris.2019.00054, arXiv : 1905.06004.
- [165] Qin WANG, Gabriel MICHAU et Olga FINK, « Missing-Class-Robust Domain Adaptation by Unilateral Alignment », in : *IEEE Transactions on Industrial Electronics* 68.1 (2021), p. 663-671, ISSN : 15579948, DOI : 10.1109/TIE.2019.2962438, arXiv : arXiv:2001.02015v1.
- [166] Qin WANG, Cees TAAL et Olga FINK, « Integrating Expert Knowledge with Domain Adaptation for Unsupervised Fault Diagnosis », in : (2021), ISSN : 15579662, DOI : 10.1109/TIM.2021.3127654, arXiv : 2107.01849, URL : <http://arxiv.org/abs/2107.01849>.
- [167] Siwei WANG, Miaomiao LI, Ning HU, En ZHU, Jingtao HU, Xinwang LIU et Jianping YIN, « K-Means Clustering With Incomplete Data », in : *IEEE Access* 7 (2019), p. 69162-69171, ISSN : 21693536, DOI : 10.1109/ACCESS.2019.2910287.
- [168] Mike WEST, « Approximating Posterior Distributions by Mixtures », in : *Journal of the Royal Statistical Society : Series B (Methodological)* 55.2 (1993), p. 409-422, DOI : 10.1111/j.2517-6161.1993.tb01911.x.
- [169] Yannick WILHELM, Peter REIMANN, Wolfgang GAUCHEL et Bernhard MITSCHANG, « Overview on hybrid approaches to fault detection and diagnosis : Combining data-driven, physics-based and knowledge-based models », in : *Procedia CIRP* 99 (2021), 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, p. 278-283, ISSN : 2212-8271, DOI : <https://doi.org/10.1016/j.procir.2021.03.041>, URL : <https://www.sciencedirect.com/science/article/pii/S2212827121003152>.
- [170] Jason L WILLIAMS, « Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter », mém. de mast., 2003, p. 247, ISBN : 9781423502616, URL : <https://apps.dtic.mil/sti/citations/ADA415317>.

- 
- [171] Jason L. WILLIAMS et Peter S. MAYBECK, « Cost-function-based hypothesis control techniques for multiple hypothesis tracking », in : *Mathematical and Computer Modelling* 43.9-10 (2006), p. 976-989, ISSN : 08957177, DOI : 10.1016/j.mcm.2005.05.022.
- [172] Yuan XIE, Jisheng ZHAO, Baohua QIANG, Luzhong MI, Chenghua TANG et Longge LI, « Attention mechanism-based CNN-LSTM model for wind turbine fault prediction using SSN ontology annotation », in : *Wireless Communications and Mobile Computing* 2021 (2021), ISSN : 15308677, DOI : 10.1155/2021/6627588.
- [173] Feixiang XU, Xinhui LIU, Wei CHEN, Chen ZHOU et Bingwei CAO, « Ontology-based method for fault diagnosis of loaders », in : *Sensors (Switzerland)* 18.3 (2018), ISSN : 14248220, DOI : 10.3390/s18030729.
- [174] Feng YANG, Xinyi TANG, Tiancheng LI et Litao ZHENG, « Learning-based Gaussian Mixture Reduction for Distributed Bayesian Filter », in : *10th International Conference on Control, Automation and Information Sciences, ICCAIS 2021 - Proceedings* (2021), p. 782-787, DOI : 10.1109/ICCAIS52680.2021.9624601.
- [175] Naoya YOKOYAMA, « An Efficient Gaussian Mixture Reduction to Two Components », in : *SASIMI 2016 Proceedings*, 2016, p. 236-241.
- [176] Lei YU, Tianyu YANG et Antoni B. CHAN, « Density-Preserving Hierarchical EM Algorithm : Simplifying Gaussian Mixture Models for Approximate Inference », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.6 (2019), p. 1323-1337, ISSN : 19393539, DOI : 10.1109/TPAMI.2018.2845371.
- [177] Liangwei ZHANG, Jing LIN, Haidong SHAO, Zhicong ZHANG, Xiaohui YAN et Jianyu LONG, « End-to-end unsupervised fault detection using a flow-based model », in : *Reliability Engineering & System Safety* 215 (2021), p. 107805, ISSN : 0951-8320, DOI : <https://doi.org/10.1016/j.res.2021.107805>, URL : <https://www.sciencedirect.com/science/article/pii/S0951832021003288>.
- [178] Qiong ZHANG, Archer Gong ZHANG et Jiahua CHEN, « Gaussian Mixture Reduction with Composite Transportation Divergence », in : *arXiv*, 2020, DOI : 10.48550/ARXIV.2002.08410, URL : <https://arxiv.org/abs/2002.08410>.
- [179] Huangjie ZHENG, Yuchen WANG, Chen HAN, Fangjie LE, Ruan HE et Jialiang LU, « Learning and Applying Ontology for Machine Learning in Cyber Attack Detection », in : *Proceedings - 17th IEEE International Conference on Trust, Se-*

---

*curity and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018* (2018), p. 1309-1315, DOI : 10.1109/TrustCom/BigDataSE.2018.00180.

- [180] Zhiwang ZHONG, Tianhua XU, Feng WANG et Tao TANG, « Text Case-Based Reasoning Framework for Fault Diagnosis and Predication by Cloud Computing », in : *Mathematical Problems in Engineering* 2018 (juill. 2018), p. 1-10, DOI : 10.1155/2018/9464971.
- [181] Qiang ZHOU, Ping YAN, Huayi LIU et Yang XIN, « A hybrid fault diagnosis method for mechanical components based on ontology and signal analysis », in : *Journal of Intelligent Manufacturing* 30.4 (2019), p. 1693-1715, ISSN : 15728145, DOI : 10.1007/s10845-017-1351-1.





## Résultats de la généralisation du chapitre 5

Dans cette section, le reste des résultats de la généralisation présentée dans le chapitre 5 sont fournis. En effet, les différentes métriques récoltés sur les 3 jeux de données (i.e. Hydro-Hydro, Hydro-Leakage, Hydro-Valve) sont présentées. L'interprétation des résultats est la même que celle décrite dans la section 5.3.3.

### Généralisation ponctuelle

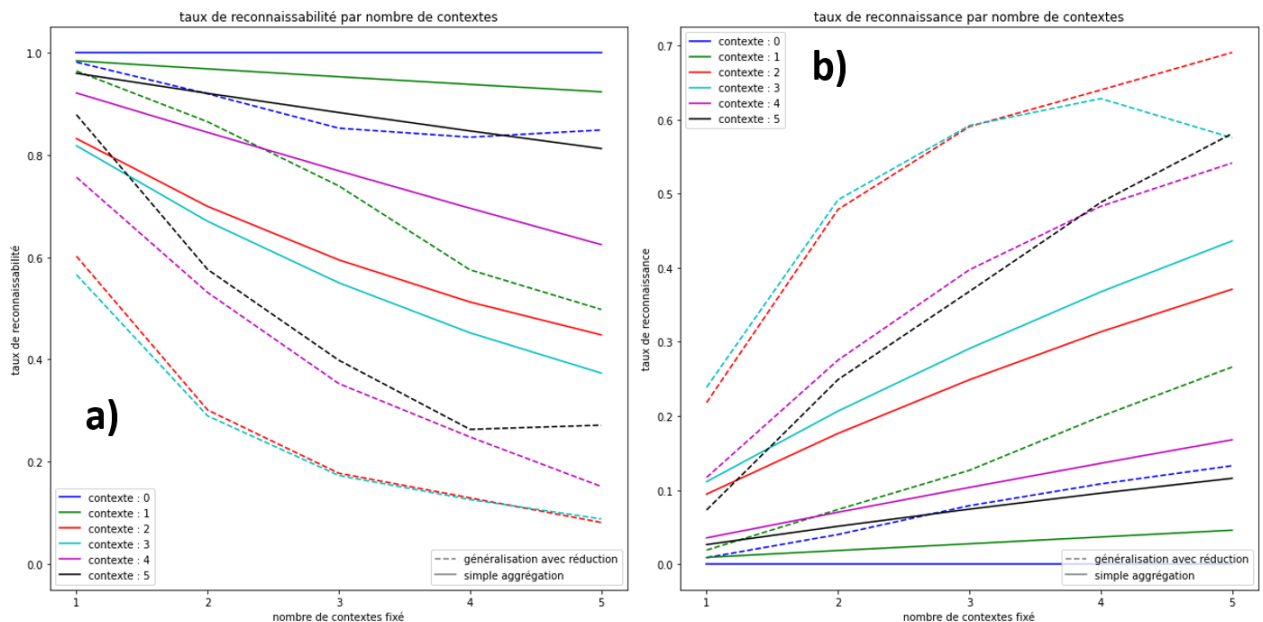


FIGURE A.1 – Résultats de généralisation ponctuelle par nombre de contexte sur Hydro Hydro : a) taux de reconnaissabilité b) taux de reconnaissance

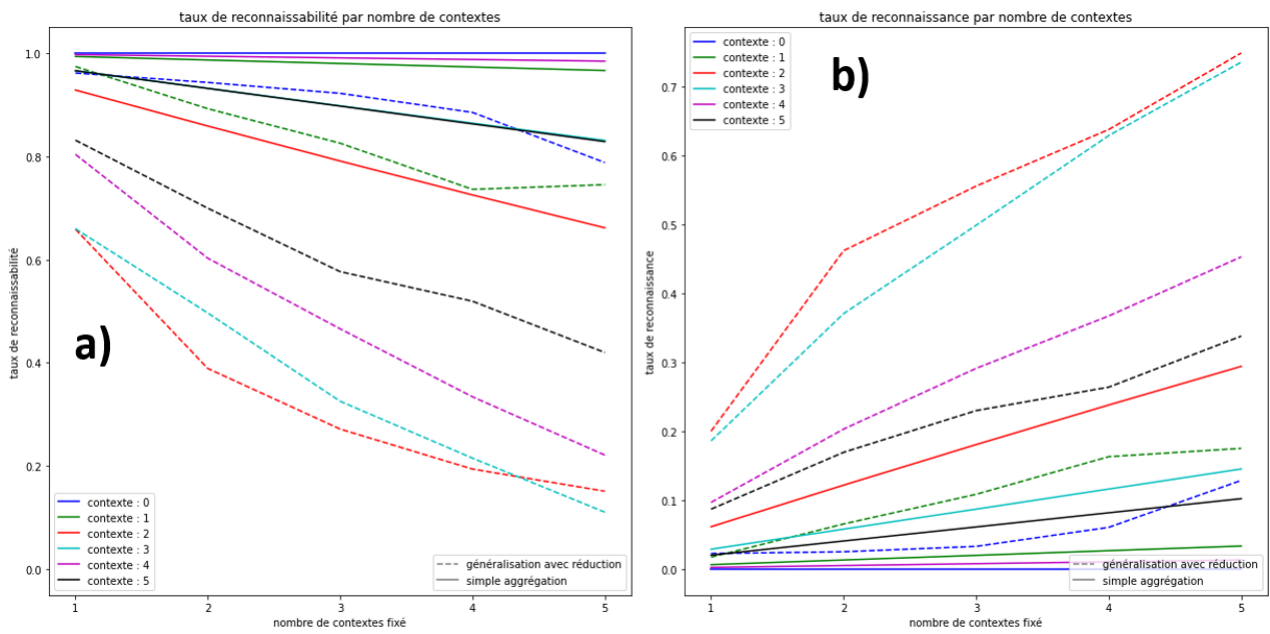


FIGURE A.2 – Résultats de généralisation ponctuelle par nombre de contexte sur Hydro Leakage : a) taux de reconnaissabilité b) taux de reconnaissance

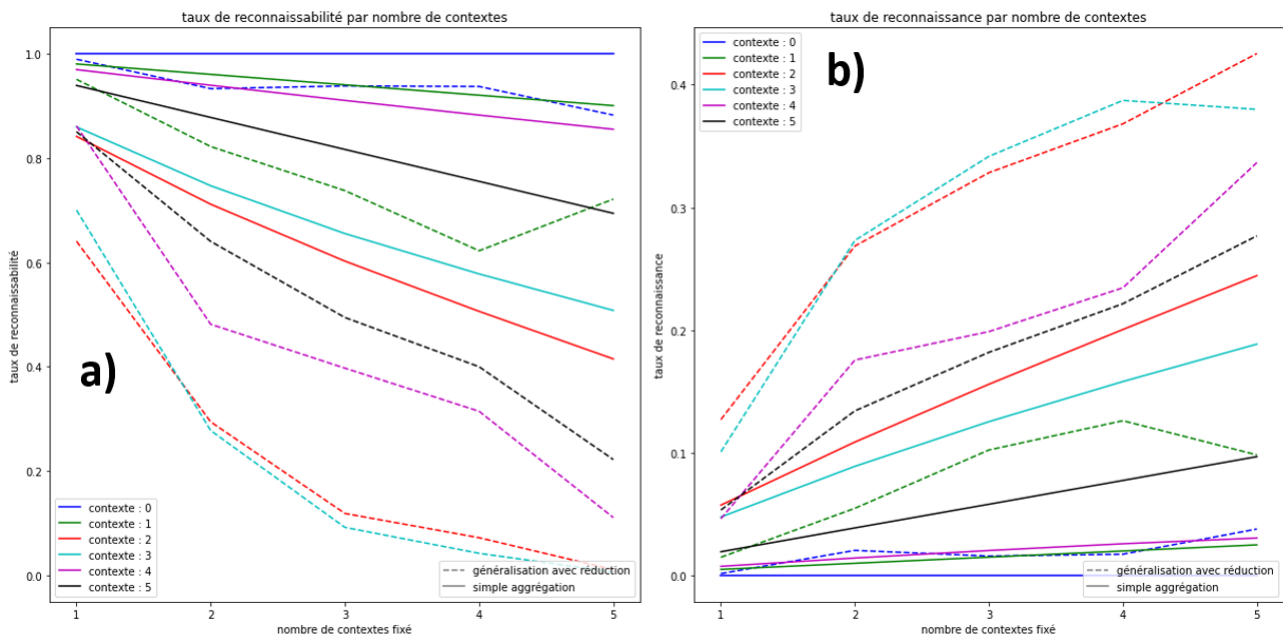


FIGURE A.3 – Résultats de généralisation ponctuelle par nombre de contexte sur Hydro Valve : a) taux de reconnaissabilité b) taux de reconnaissance

## Généralisation appliquée à l'apprentissage itératif

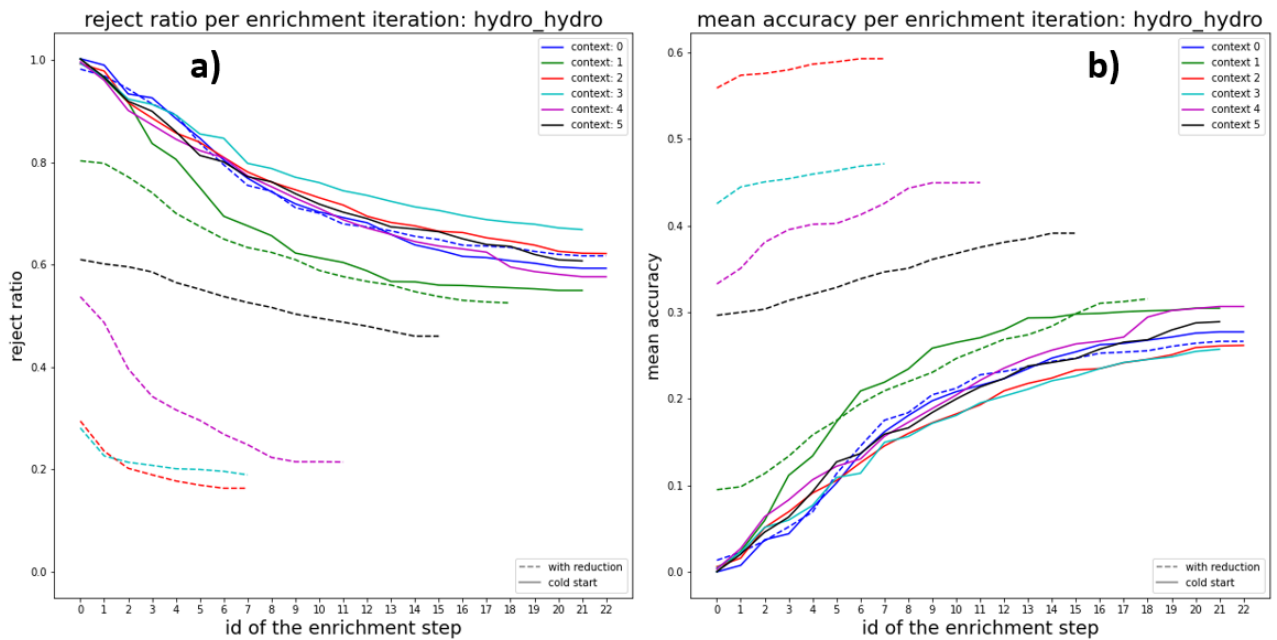


FIGURE A.4 – Résultats de généralisation appliquée à l'apprentissage itératif sur Hydro Hydro : a) taux de reconnaissabilité b) taux de reconnaissance

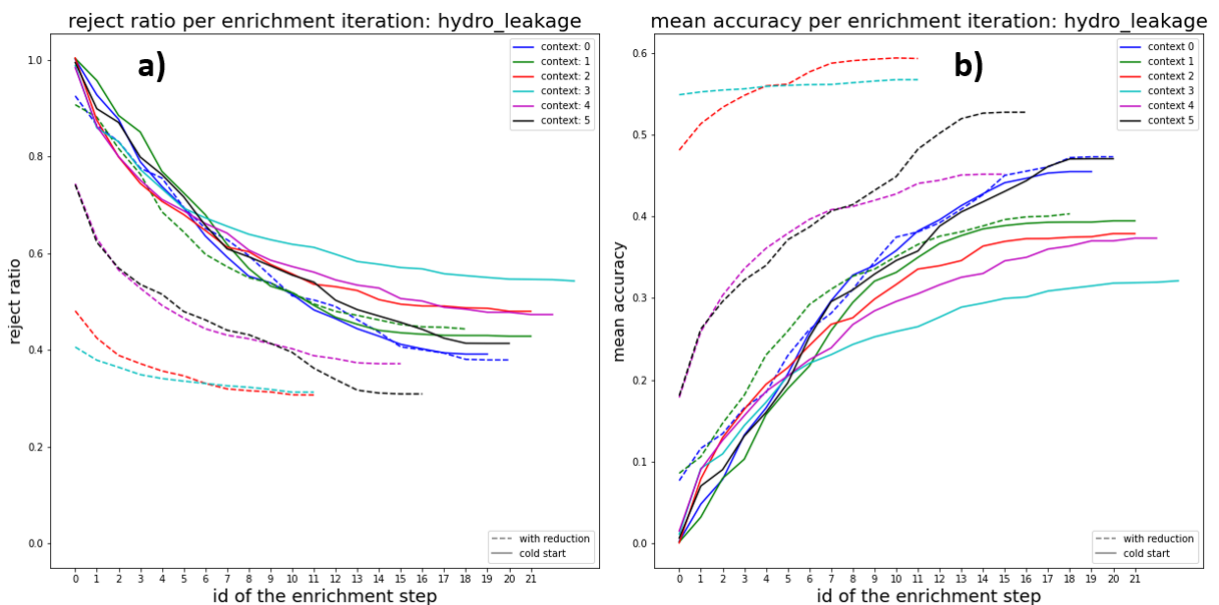


FIGURE A.5 – Résultats de généralisation appliquée à l'apprentissage itératif sur Hydro Leakage : a) taux de reconnaissabilité b) taux de reconnaissance

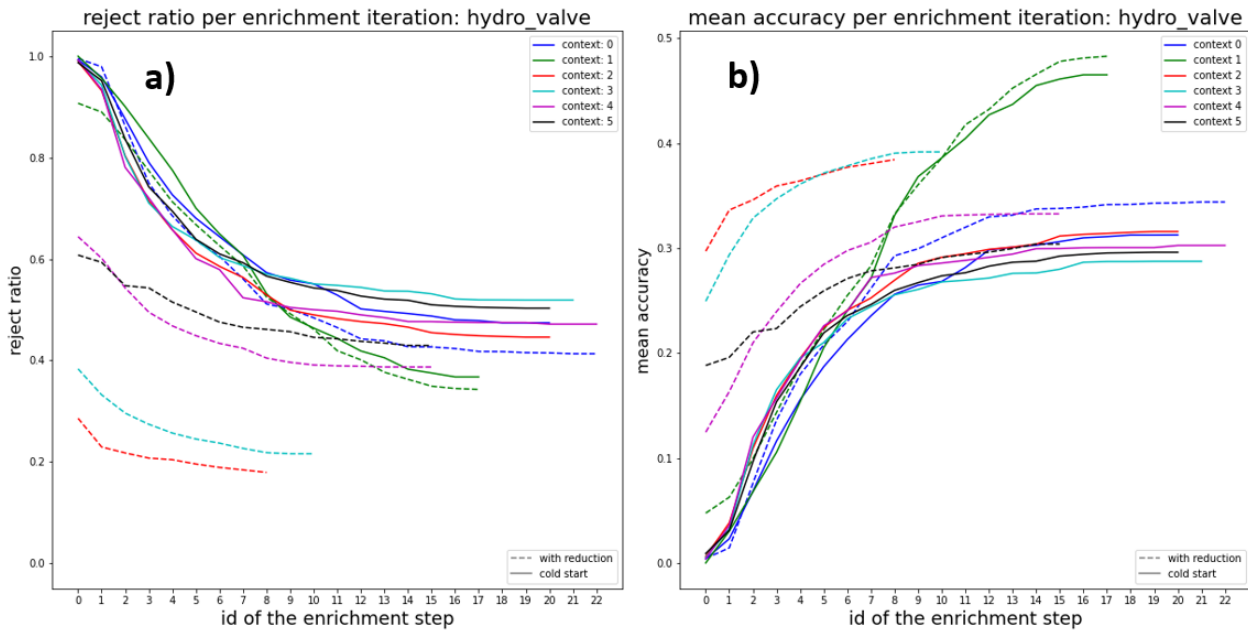


FIGURE A.6 – Résultats de généralisation appliquée à l'apprentissage itératif sur Hydro Valve : a) taux de reconnaissabilité b) taux de reconnaissance

## Courbes de vissage

Dans cette section, les courbes de vissage utilisées pour les expériences du chapitre 6 sont présentées. En effet, les courbes originales ainsi que celles augmentées sur la classe minoritaires sont illustrées. Il convient de rappeler que seule la partie de montée en couple

est affichée (cf. figure 6.2).

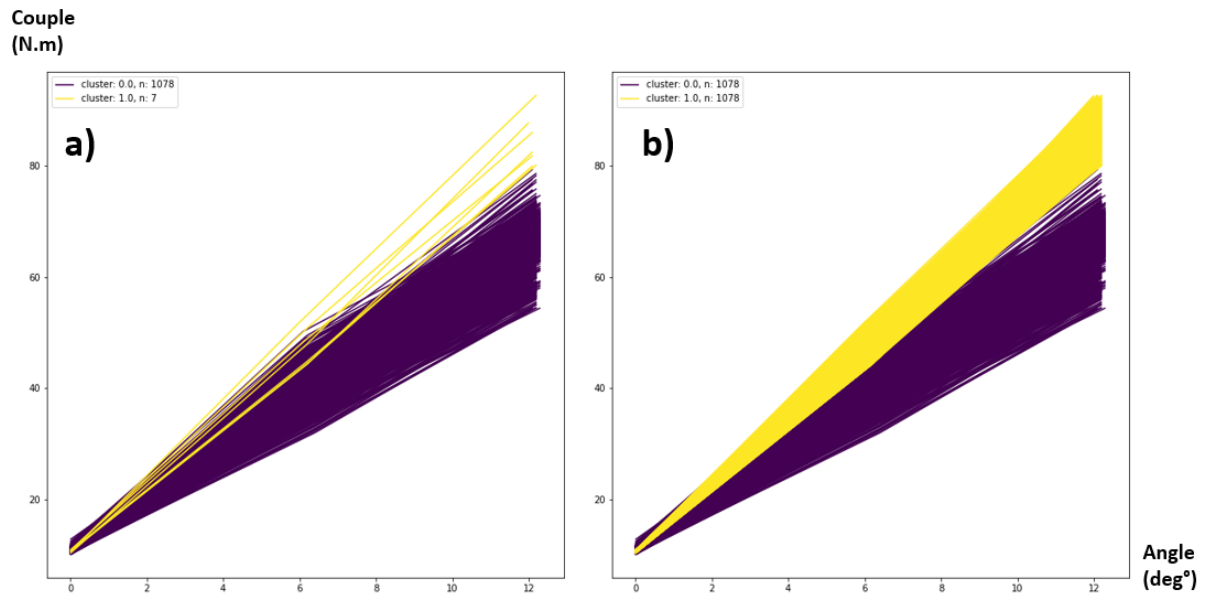


FIGURE A.7 – Courbes de vissage du contexte 0 : a) courbes originales b) courbes après l'augmentation des courbes de la classe minoritaire

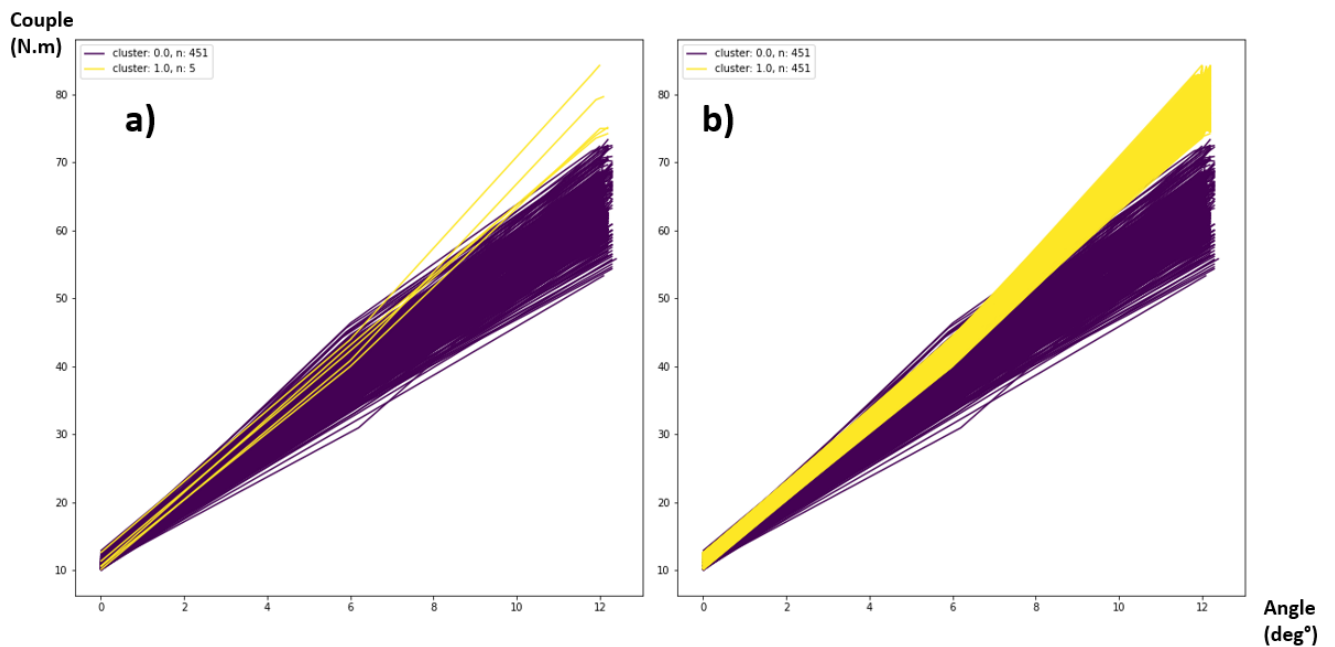


FIGURE A.8 – Courbes de vissage du contexte 1 : a) courbes originales b) courbes après l'augmentation des courbes de la classe minoritaire

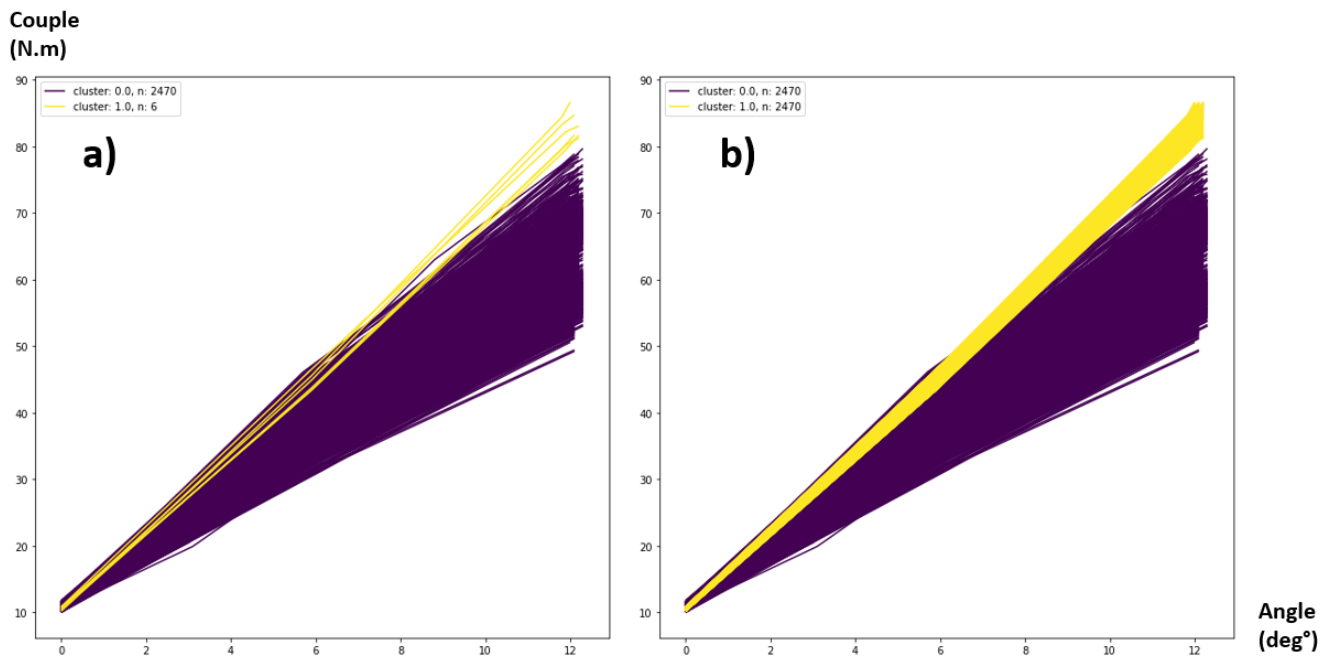


FIGURE A.9 – Courbes de vissage du contexte 2 : a) courbes originales b) courbes après l’augmentation des courbes de la classe minoritaire

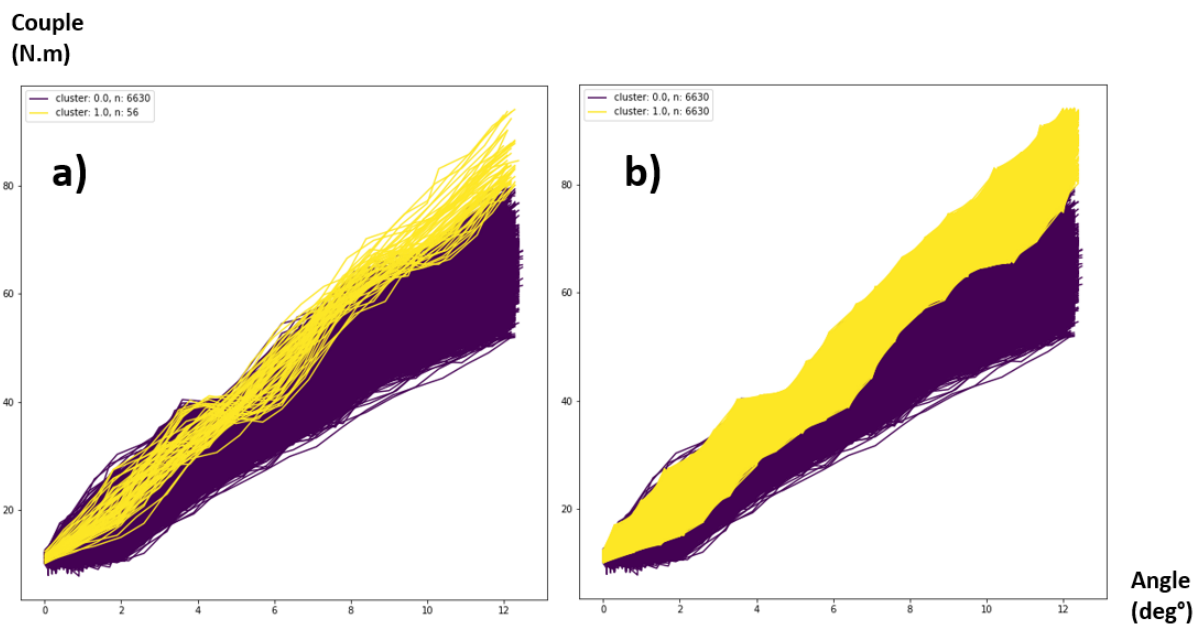


FIGURE A.10 – Courbes de vissage du contexte 3 : a) courbes originales b) courbes après l’augmentation des courbes de la classe minoritaire

Couple  
(N.m)

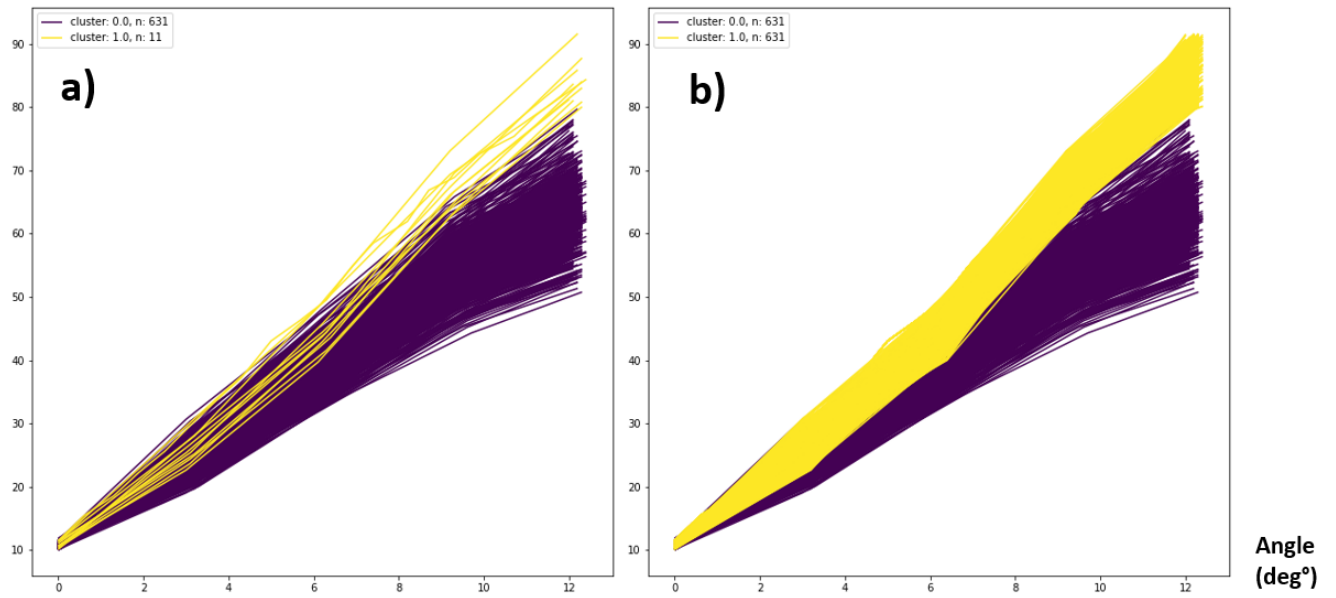


FIGURE A.11 – Courbes de vissage du contexte 4 : a) courbes originales b) courbes après l'augmentation des courbes de la classe minoritaire







---

**Titre :** Découverte et Généralisation des Connaissances dans les Procédés de Fabrication

**Mot clés :** Apprentissage automatique, Base de connaissances, Détection de défauts

**Résumé :** L'objectif de cette thèse est de proposer une nouvelle architecture dédiée à la détection de défauts dans les procédés de fabrication. Cette dernière repose sur la combinaison des techniques d'apprentissage automatique (ML) avec une Ontologie générique. L'architecture est évolutive dans le sens où de nouveaux défauts peuvent être découverts par le ML et ajoutés en tant que nouvelles connaissances dans l'ontologie de façon périodique. Cela facilite la détection des défauts dans les contextes industriels, où les défauts sont généralement rares. D'autre part, l'inspection périodique des connaissances capturées itérativement par l'Ontologie nous permet de définir des connaissances plus gé-

nériques. Celles-ci permettront dans un premier temps d'améliorer la détection de défauts dans des contextes connus et puis, dans un second temps, de reconnaître les défauts connus dans de nouveaux contextes industriels. Ces tâches sont liées aux domaines de l'apprentissage par transfert et de l'adaptation au domaine, appliquées ici pour induire de nouvelles connaissances dans l'Ontologie, et augmenter les performances du système de détection de défauts. Des expériences sur des données UCI ainsi que sur des données issues des opérations de vissage réelles nous ont permis de valider la fiabilité et la robustesse de l'approche.

---

**Title:** Discovery and Generalization of Knowledge in Manufacturing Processes

**Keywords:** Machine Learning, Knowledge Base, Fault detection

**Abstract:** The objective of this thesis is to propose a novel architecture dedicated to fault detection in manufacturing processes. It is based on the combination of Machine Learning (ML) techniques with a generic Ontology. The architecture is evolving in the sense that new defects can be discovered by the ML and added as new knowledge in the ontology periodically. This facilitates defect detection in industrial contexts, where defects are generally rare. On the other hand, the periodic inspection of the knowledge captured iteratively by the Ontology allows us to define more generic

knowledge. This knowledge will first allow us to improve the detection of defects in known contexts and then, in a second step, to recognize known defects in new industrial contexts. These tasks are related to the fields of transfer learning and domain adaptation, applied here to induce new knowledge in the Ontology, and to increase the performances of the fault detection system. Experiments on several UCI datasets as well as on data from real screwing operations allowed us to validate the reliability and robustness of the approach.