Modèles graphiques probabilistes appliqués aux procédés de fabrication

Mathilde Monvoisin

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

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
Mathilde MONVOISIN

Modèles graphiques probabilistes appliqués aux procédés de fabrication

Thèse présentée et soutenue à Nantes, le 16/12/2022
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Philippe WEBER — Professeur à l'Université de Lorraine
Karim TABIA — Maître de conférence HDR à l'Université d'Artois

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président : Bruno CASTANIER	Professeur à l'Université d'Angers	
	Philippe WEBER	Professeur à l'Université de Lorraine
	Karim TABIA	Maître de conférence HDR à l'Université d'Artois
	Véronique DELCROIX	Maître de conférence à l'Université Polytechnique Hauts-de-
Dir. de thèse :	Philippe LERAY	Professeur à Nantes Université
Co-dir. de thèse :	Mathieu RITOU	Maître de conférence HDR à Nantes Université

Invité :

Adolfo SUAREZ ROOS — Expert R&T IRT Jules Verne

Remerciements

Je souhaiterais tout d'abord remercier mes directeurs et co-directeurs de thèse Philippe Leray et Mathieu Ritou. Le doctorat n'était pas un long fleuve tranquille mais c'est grâce à vous que j'en ressors une expérience extrêmement enrichissante et positive. J'ai énormément appris à vos côtés.

Je remercie également tous les collègues aux côtés desquels j'ai vécu cette aventure, tous les doctorants des équipes DUKe et IPI, et particulièrement Tristan, Sarah et bien sûr Mokhtar, merci de m'avoir soutenue au quotidien à Polytech et chez nous. Merci pour toutes nos conversations et ta positivité.

Merci à mes amis pour votre soutien et votre enthousiasme : Nathan, Paule, Aurore, Dorian, Ambre, Silke, Laurène, Laura, Thomas, Lucas. Merci à Pamplemousse, Kiwi et Picodon pour leur présence et leur oreille attentive.

Je souhaite remercier tous les membres de ma famille, Joël, ma petite soeur Camille et Isabelle. Par dessus tout, je tiens à remercier mes parents d'avoir toujours été des modèles de persévérance et d'éthique. Je suis reconnaissante à ma mère d'avoir toujours été présente et de m'avoir appris à ne pas baisser les bras. C'est ta force qui m'a permis d'arriver là où j'en suis. Enfin, bien que tu ne sois plus à mes côtés aujourd'hui, je remercie mon père pour ses enseignements passés. Merci pour ton courage et de m'avoir convaincue que tout est possible.

TABLE DES MATIÈRES

INTRODUCTION

Contexte du travail et motivations

Le travail réalisé se situe dans le cadre de l'industrie 4.0, avec un intérêt particulier pour l'usinage à grande vitesse. Les modèles graphiques probabilistes utilisés sont les réseaux bayésiens, qui ont déjà une place dans le diagnostic industriel, mais dont l'apprentissage des paramètres avec le co-training non-supervisé n'est pas exploré.

L'industrie 4.0

La fabrication intelligente est un domaine de recherche prometteur pour l'amélioration de la productivité et de la compétitivité dans l'industrie, en exploitant les données numériques obtenues lors de procédés de fabrication [119, 130]. En effet, il est crucial de détecter toute défaillance du système le plus tôt possible afin de réduire les coûts de maintenance et les temps d'arrêt. L'utilisation de systèmes intelligents peut permettre de réduire la durée des arrêts grâce à des détections rapides, ou encore de les éviter grâce à des méthodes de prédiction. C'est la raison pour laquelle la maintenance prédictive est un enjeu essentiel de l'industrie 4.0 [40]. Ces approches peuvent être basées sur des données, sur des modèles ou hybrides.

La fabrication intelligente peut utiliser l'apprentissage supervisé afin de construire un modèle prédictif, comme un réseau neuronal probabiliste pour classer les outils cassés et les bons outils [56] ou les réseaux bayésiens et les machines à vecteurs de support pour la modélisation et la prédiction thermique [107].

Cette thèse a pu être réalisée grâce à l'Institut de Recherche Technologique (IRT) Jules Verne, dans le cadre du programme de thèses PERFORM, dont le but est de stimuler la recherche fondamentale dans le domaine de la fabrication. Le travail présenté dans ce manuscrit se trouve à la croisée entre l'expertise en fabrication de l'équipe RoMaS (Robots and Machines for Manufacturing, Society and Services) du LS2N et celle de l'équipe DUKe en construction de modèles de connaissances par les données centrées sur l'utilisateur, c'est-à-dire adaptables et interprétables.

Problématiques spécifiques à l'usinage

La maintenance conditionnelle des machines tournantes est généralement basée sur des mesures périodiques des vibrations [34, 83]. Des analyses d'huile [24] et des mesures thermiques peuvent également être effectuées. En pratique, l'analyse du signal vibratoire des machines tournantes peut être réalisée à l'aide de critères globaux qui évaluent le niveau de vibration, tels que la valeur efficace de vitesse d'oscillation (V_{rms}). Elle est définie dans la norme ISO 10816-3 [59] et elle est largement utilisée pour la maintenance conditionnelle. La norme ISO 17243-1 [60] recommande d'utiliser à la fois la vitesse V_{rms} et l'accélération A_{rms} pour la surveillance de l'état de la broche. Pour une surveillance plus précise, le spectre de fréquence des vibrations peut être analysé. En particulier, *BPFO* et *BPFI* permettent de révéler un défaut local sur la bague extérieure ou la bague intérieure d'un roulement respectivement. En outre, les défauts distribués peuvent également être surveillés, comme l'ondulation de la bague [126] ou l'aggravation de la rugosité de surface [27]. Randall et Antoni [108] présentent l'état de l'art pour le diagnostic des roulements par l'analyse du signal vibratoire. Jimenez et al [64] passent en revue les articles des dix dernières années traitant de la maintenance prédictive, dont l'objectif est de prédire la durée de vie utile restante.

La surveillance du procédé d'usinage consiste à analyser les signaux en cours d'usinage, afin de détecter les bris d'outils, ou de prédire la qualité de surface de la pièce usinée. La plupart des recherches sur la surveillance des conditions d'usinage se concentrent sur la surveillance de l'usure de l'outil [120]. D'autres articles considèrent l'impact du broutement sur la qualité de surface de la surface usinée [103]. La littérature traite de la

prédiction et de la surveillance de ces vibrations instables, par des signaux de vibration ou de force [92, 74]. La surveillance de l'état des outils peut s'appuyer sur l'apprentissage automatique, généralement de manière supervisée, pour construire un modèle prédictif [69]. Par exemple, un réseau neuronal probabiliste peut classer les outils cassés et ceux en bon état [56].

Réseaux bayésiens et utilisation dans l'industrie

Les réseaux bayésiens (RB) sont des modèles graphiques probabilistes adaptés à la maintenance ou à la surveillance en cours de processus dans l'industrie manufacturière [135]. En effet, leur capacité à modéliser des systèmes complexes sous incertitude, à apprendre ces modèles avec des jeux de données non étiquetés, incomplets, déséquilibrés ou de petite taille [9] est bien connue. Tidriri et al [121] passent en revue les principales approches guidées par les données et basées sur les modèles pour le diagnostic des défauts de processus et comparent leurs avantages et leurs inconvénients respectifs. Les réseaux bayésiens offrent une grande fiabilité pour gérer l'incertitude. Par exemple, Atoui et al [5] proposent d'utiliser des réseaux bayésiens avec un modèle de mélange gaussien pour la surveillance supervisée de procédés de fabrication guidée par les données. Des approches supervisées sont souvent proposées, alors que la collecte de données étiquetées en cours de processus est généralement impossible dans l'industrie. De plus, l'interaction potentielle entre l'état de la machine et l'état du procédé n'est pas étudiée dans la littérature, et encore moins avec la prise en compte des incertitudes.

Co-training

Améliorer l'apprentissage d'un modèle donné avec les résultats d'un autre modèle est l'un des principes fondateurs du co-training. Blum et Mitchell [13] et Nigam et Ghani [96] ont proposé un paradigme d'apprentissage semi-supervisé, qui entraîne deux classifieurs bayésiens naïfs à partir de deux vues différentes et laisse les classifieurs étiqueter des données non étiquetées l'un pour l'autre. Yu et al [137] a utilisé un modèle graphique bayésien non dirigé pour le co-training.

L'apprentissage avec le co-training se fait classiquement de façon semi-supervisée. Le travail de cette thèse sur l'apprentissage des paramètres avec le co-training se concentre sur le développement de stratégies non-supervisées.

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

La partie du manuscrit sur l'état de l'art comporte le chapitre 1 sur les réseaux bayésiens, qui développe la définition formelle de ces modèles, les méthodes d'apprentissage de leur structure et de leurs paramètres, en particulier dans le cas de données manquantes. Ce chapitre décrit également des concepts spécifiques à notre problématique, tels que l'utilisation de fonctions d'agrégation, les méthodes de discrétisation de variables ou encore la sélection de variables. Le chapitre 2 sur le co-training présente les principes et limitations du co-training. Les hypothèses intrinsèques aux techniques de co-training y sont détaillées ainsi qu'une discussion sur l'importance de ces hypothèses.

Contributions

La partie contributions de la thèse est constituée de trois chapitres qui englobent deux thèmes majeurs de recherche et leurs applications. Le chapitre 3 présente une structure générique de réseau bayésien pour le diagnostic (industriel, médical, etc) à partir d'un ensemble de capteurs. Ce chapitre détaille également la méthode d'apprentissage de ce modèle et la stratégie d'utilisation de cette architecture pour la sélection des entrées pertinentes pour le diagnostic. La méthode de sélection des variables est validée sur plusieurs jeux de données artificiels.

Le chapitre 4 développe les avancées de cette thèse dans le domaine du co-training non-supervisé, avec pour contributions trois stratégies génériques utilisables par tout modèle graphique probabiliste. Les performances de ces stratégies de co-training non-supervisées sont validées sur un ensemble de jeux de données artificiels et de jeux de données issus du dépôt de référence UCI [33].

Comme le montre la figure 1, les contributions des deux premiers chapitres sont alors reprises dans le chapitre 5 qui détaille l'utilisation de deux réseaux bayésiens, dont les paramètres sont appris grâce à nos stratégies de co-training non-supervisé sur un jeu de données d'usinage issu de conditions réelles de fabrication. La méthode de sélection des variables évoquée précédemment est appliquée sur ce jeu de données.

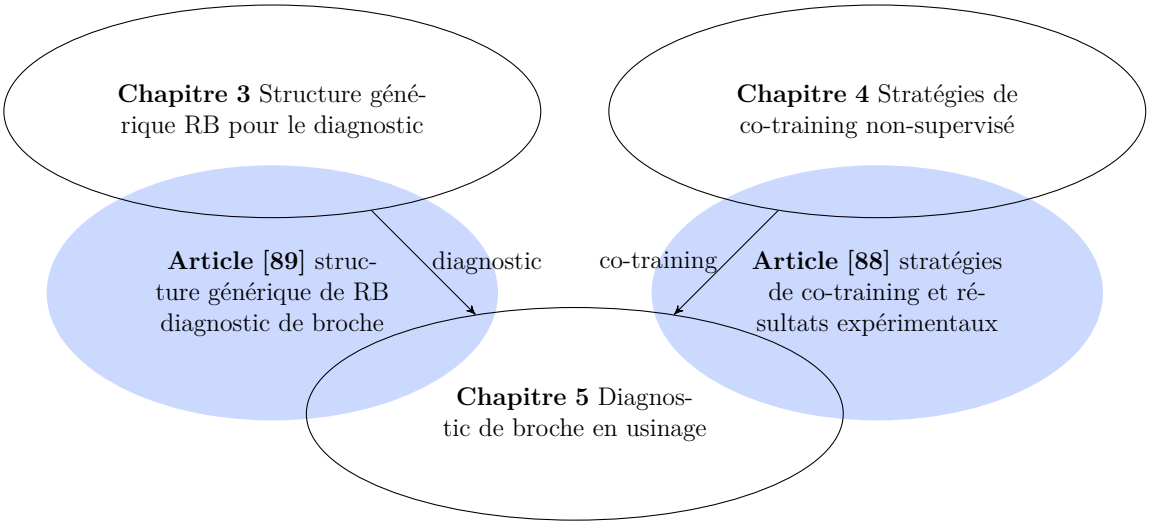


FIGURE 1 – Organisation des contributions du manuscrit : chapitres et publications

La structure générique détaillée dans le chapitre 3 a été originellement publiée dans l'article [89]. Les propositions de l'article [88] se situent quant à elles dans les stratégies de co-training non supervisées du chapitre 4.

Table de notations

La table 2 reprend l'ensemble des notations utilisées dans le manuscrit en deux parties distinctes : les notations liées aux données, et celles liées au réseau bayésien.

Notation	Objet en lien avec les données
$\mathcal{D} = \{\mathcal{D}_o, \mathcal{D}_m\}$	ensemble des données, composé de $\mathcal{D}_o$ et $\mathcal{D}_m$ respectivement les données observées et les données manquantes
$d = \{d_o, d_m\}$	instance des données, composé de d_o et d_m respectivement les valeurs observées et les valeurs manquantes
y	valeur de la vérité de la classe pour une instance donnée
$\mathcal{D}_\mathcal{A}$	ensemble des données annotées
$\mathcal{D}_{\bar{\mathcal{A}}}$	ensemble des données non-annotées
$d_A \in \mathcal{D}_\mathcal{A}$	instance annotée
$d_{\bar{A}} \in \mathcal{D}_{\bar{\mathcal{A}}}$	instance non-annotée
suite à la page suivante...	

suite de la table de notations	
$\mathcal{D}^{\mathcal{C}}$	ensemble des données classifiées (originellement non-annotées mais dont une étiquette a été ajoutée par un classifieur)
r_D	ratio de déséquilibre des données (taux d'instances positives)
N	nombre d'échantillons
$\mathcal{X}_i$	$i$ème vue (co-training)
Notation	Objet en lien avec le réseau bayésien
$\mathcal{M}$	le modèle
$\mathcal{G}$	graphe
$\mathbf{G}$	espace des graphes
D	distribution
θ	paramètres du RB
X	variable
x	valeur prise par une variable
$\mathbf{X}$	ensemble de variables
$\mathbf{x}$	ensemble de valeurs de variables
n	nombre de variables
d	nombre de paramètres libres du modèle
r_i	nombre d'états de la variable X_i
$pa(X_i)$	parents de X_i dans la structure du réseau bayésien
l	prédiction en sortie du modèle pour une instance donnée
q_i	nombre d'états de $Pa(X_i)$
θ_{ijk}	$= P(X_i = x_k \vert Pa(X_i) = x_j)$
N'_{ijk}	exposant de Dirichlet de θ_{ijk}
N'_{ij}	$= \sum_{k=1}^{r_i} N'_{ijk}$
N_{ijk}	nombre d'entrées dans un jeu de données où $X_i = x_k$ et $Pa(X_i) = x_j$
N_{ij}	$= \sum_{k=1}^{r_i} N_{ijk}$
α	*a priori* de Dirichlet
α_{ijk}	*a priori* de Dirichlet pour $X_i = x_k$ et $Pa(X_i) = x_j$
H	entropie
$\tau_{j,i}$	$j$ème seuil du $i$ème capteur

TABLE 2 – Table des notations

Première partie

État de l'art

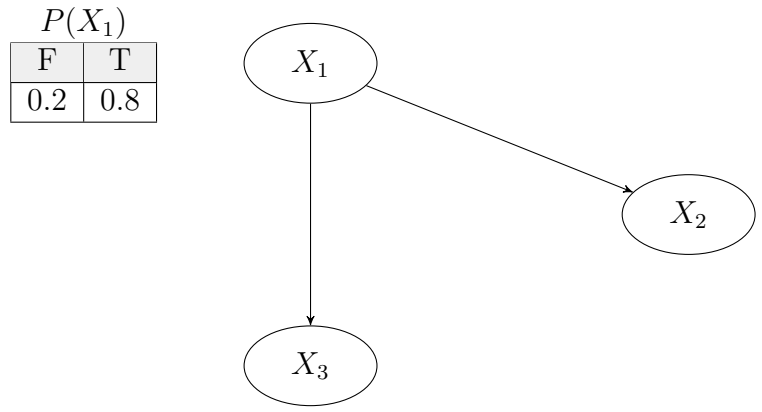
Chapitre 1

Réseaux bayésiens

Les réseaux bayésiens [101, 16] sont une famille de modèles probabilistes qui sont capables de représenter de manière concise les relations de dépendances entre les variables. La partie 1.1 rappelle la définition formelle d'un réseau bayésien. La partie 1.2 évoque les différentes méthodes existantes pour l'apprentissage de la structure et/ou des paramètres dans un réseaux bayésien. La partie 1.3 détaille le cas des variables continues. La partie 1.4 passe en revue les possibles utilisations des réseaux bayésiens pour la classification et/ou le diagnostic.

1.1 Définition

Definition 1.1.1. *Un réseau bayésien (RB) [16] est défini par un graphe $\mathcal{G}$ et ses paramètres θ tels que :*



$P(X_1)$

F	T
0.2	0.8

$P(X_2|X_1)$

X_1	F	T
F	0.4	0.6
T	0.01	0.99

$P(X_3|X_1)$

X_1	F	T
F	0.4	0.6
T	0.2	0.8

FIGURE 1.1 – Exemple de réseau bayésien

— $\mathcal{G} = (V, E)$ *est un graphe orienté sans circuit où* V *est un ensemble de variables aléatoires* $(X_1, X_2, ..., X_n)$ *qui correspondent aux nœuds du graphe* $\mathcal{G}$ *et* E *est un ensemble d'arêtes ;*

— $\theta = \{P(X_i|Pa(X_i))\}$ *est l'ensemble des distributions de probabilités conditionnelles de chaque nœud sachant ses parents, où* $Pa(X_i)$ *sont les parents de* X_i *dans le graphe* $\mathcal{G}$*.*

La figure 1.1 montre un réseau bayésien avec trois variables aléatoires et leurs distributions de probabilité conditionnelles correspondantes. Dans un souci de simplification, chacune des variables aléatoires de cet exemple est booléenne et ne peut prendre que deux valeurs, Vrai (T) ou Faux (F), mais en pratique, il n'y a aucune restriction sur le nombre de valeurs que peut prendre une variable aléatoire discrète.

Tout modèle probabiliste complet doit, explicitement ou implicitement, représenter la distribution conjointe, c'est-à-dire la probabilité de chaque événement possible tel que défini par les valeurs de toutes les variables. Les réseaux bayésiens sont compacts car ils factorisent la distribution conjointe en distributions conditionnelles locales pour chaque variable compte tenu de ses parents. En outre, toute distribution conjointe sur un ensemble de variables peut être représentée par un réseau bayésien entièrement connecté.

L'interprétation générale des réseaux bayésiens indique que la distribution jointe s'écrit comme dans l'équation 1.1.

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)) \tag{1.1}$$

Les réseaux bayésiens constituent un moyen approprié pour aborder de nombreux problèmes d'intelligence artificielle, si l'objectif est de parvenir à des conclusions de manière probabiliste plutôt que sous forme de propriétés logiques.

1.2 Apprentissage de réseaux bayésiens

Les réseaux bayésiens, de par leur constitution, présentent l'avantage de pouvoir être défini à la main à partir de connaissances préalables, ou à travers un algorithme d'apprentissage à partir de données. Il existe de nombreux algorithmes pour apprendre la structure et/ou les paramètres d'un réseau bayésien à partir de données. Les sous-parties 1.2.1 et 1.2.2 évoquent certaines méthodes existantes, fréquemment utilisées pour de l'apprentissage respectivement de structure et de paramètres. Daly et al [25] détaillent l'étendue des méthodes développées durant les dernières décennies pour l'apprentissage des réseaux bayésiens, et essentiellement de leur structure.

1.2.1 Apprentissage de structure

L'apprentissage automatique de structure consiste, à partir de l'ensemble de variables d'entrées à construire le graphe $\mathcal{G}$ en créant des arcs entre certaines variables.

Les algorithmes existants pour la construction de la structure du graphe sont répartis notamment en méthodes basées sur le score et en méthodes basées sur la contrainte.

Méthodes basées sur le score

Le but est de maximiser une fonction de score telle que décrite dans l'équation 1.2.

$$\arg\max_{\mathcal{G}\in\mathbf{G}} score(\mathcal{G}, \mathcal{D}) \tag{1.2}$$

Un des scores fréquemment utilisés est le *Bayes Dirichlet equivalent uniform* (BDeu) [15, 22, 52] qui mesure la probabilité d'un graphe dirigé sans circuit choisi compte tenu des données disponibles. Le BDeu suppose une distribution de probabilité a priori uniforme

sur tous les graphes possibles est un cas particulier du *Bayes Dirichlet* (BD) avec $N'_{ijk} = N'/r_i q_i$. Ce dernier a été tout d'abord décrit par Cooper et Herskovits [22] puis a été repris par Heckerman et Geiger [51] pour créer le BDeu, dans l'équation 1.3.

$$BDeu(\mathcal{G}, \mathcal{D}) = P(\mathcal{G}) \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ijk})}{\Gamma(N'_{ijk} + N_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk})}{\Gamma(N'_{ijk} + N_{ijk})} \tag{1.3}$$

Le score *Bayesian Information Criterion* (BIC), défini dans l'équation 1.4 est également très largement utilisé. d est le nombre de paramètres libres du modèle. Le principe de ce score est de pénaliser la complexité du modèle pour éviter le surapprentissage.

$$BIC(\mathcal{G}, \mathcal{D}) = -\log P(\mathcal{D}|\theta, \mathcal{G}) + \frac{d}{2} \log n \tag{1.4}$$

Ces deux scores ont pour but d'approcher la vraisemblance marginale $P(\mathcal{D}|\mathcal{G})$ et ont l'avantage d'être décomposables.

Méthodes basées sur la contrainte

Ces algorithmes utilisent une série de tests d'indépendance conditionnelle pour apprendre les indépendances entre les variables du modèle.

Un des algorithmes fréquemment utilisés est l'algorithme PC [116], qui part d'un graphe complet non-orienté et supprime des arcs en fonction des indépendances conditionnelles calculées. Cet algorithme peut-être utilisé avec un test du χ^2 ou d'information mutuelle (IM). L'algorithme PC classique est dépendant de l'ordre des variables et a été amélioré par Colombo et al [21] pour le rendre indépendant de l'ordre.

1.2.2 Apprentissage automatique de paramètres

L'apprentissage des distributions de probabilité à partir de données prend place après l'obtention du graphe et peut-être fait avec des données dites complètes ou incomplètes. Dans ce deuxième cas, soit il peut manquer des valeurs dans certaines lignes ou colonnes de façon éparpillée, soit il s'agit de variables sans aucune observation, *i.e.* variables latentes.

Données complètes

Maximum likelihoog estimate (MLE) est une méthode fréquemment utilisée pour l'apprentissage de paramètres avec des données complètes [62]. La formule pour le calcul des

Algorithme 1 : Espérance-maximisation

Entrée : $\mathcal{D}$, θ_0, $seuil_\epsilon$
Sortie : θ^*

$\theta = \theta_0$
tant que $\epsilon > seuil_\epsilon$ **faire**
  $Q(\theta|\theta_i) = E[\log P(\mathcal{D}|\theta)|\mathcal{D}_o, \theta_i]$ `// étape du calcul de l'espérance`
  $\theta_{i+1} = \arg\max_\theta(Q(\theta|\theta_i))$ `// étape de maximisation`
  $\epsilon =$ variation de la vraisemblance
$\theta^* = \theta_i$

probabilités se trouve dans l'équation 1.5 [44], où N_{ijk} est le nombre d'entrées dans un jeu de données où $X_i = x_k$ et $Pa(X_i) = x_j$ et $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

$$\theta_{ijk}^{MLE} = \frac{N_{ijk}}{N_{ij}} \tag{1.5}$$

La méthode d'estimation des paramètres *expected a posteriori* (EAP) est également populaire, sa formule est donnée par l'équation 1.6 [61], où α_{ijk} est l'*a priori* de Dirichlet pour $X_i = x_k$ et $Pa(X_i) = x_j$.

$$\theta_{ijk}^{EAP} = \frac{N_{ijk} + \alpha_{ijk}}{\sum_{l=1}^{r_i}(N_{ijl} + \alpha_{ijl})} \tag{1.6}$$

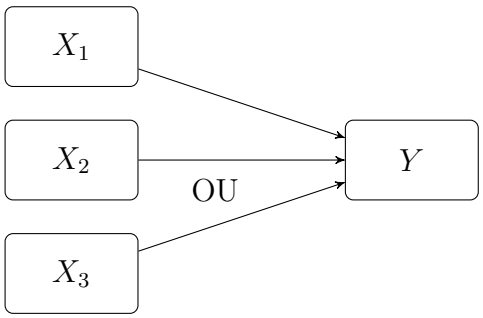
Données incomplètes

L'apprentissage avec des données incomplètes est plus délicat. Plusieurs méthodes existent pour ce faire, telles que la méthode de Monte-Carlo, l'estimation bayésienne robuste, ou encore l'algorithme *expectation-maximization* (EM), qui fait partie des plus utilisés.

EM a été introduit par Dempster et al [28] pour l'apprentissage des données incomplètes et est détaillé dans l'algorithme 1. C'est un algorithme itératif qui fonctionne par alternance de deux étapes :

1. Estimation de la probabilité des valeurs manquantes en fonction de la valeur actuelle des paramètres à estimer ;
2. Actualisation des paramètres.

La condition d'arrêt de EM peut être un nombre d'itérations maximum ou un seuil de variation de la vraisemblance.



$P(Y|X_1, X_2, X_3)$

X_1	X_2	X_3	F	T
F	F	F	1	0
F	F	T	0	1
F	T	F	0	1
F	T	T	0	1
...				
T	T	T	0	1

FIGURE 1.2 – Fonction déterministe : exemple du **ou logique**

EM présente l'inconvénient [39] d'être très dépendant de l'initialisation et de s'arrêter souvent sur des optimums locaux. Il est souvent très intéressant d'avoir une bonne connaissance du système modélisé pour initialiser de manière pertinente les paramètres afin d'obtenir de meilleures performances.

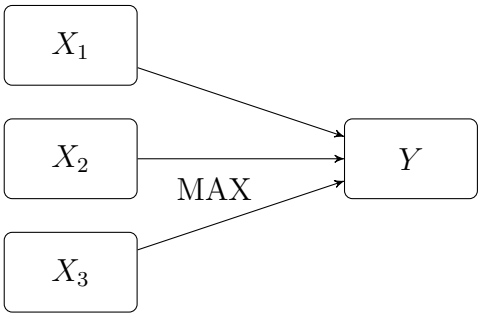
1.2.3 Formes particulières de distributions

Certaines distributions particulières de distributions sont adaptées et permettent d'améliorer la définition du modèle dans certain cas. Les fonctions déterministes et les interactions canoniques font partie de ces formes particulières.

Fonctions déterministes

Utiliser une fonction déterministe pour définir la table de probabilité conditionnelle d'une variable a plusieurs avantages. Cela permet de gagner en temps de calcul car il n'y a pas calculs à effectuer pour estimer la distribution conditionnelle pour cette dépendance. Pour un modèle difficile à apprendre (peu de données à disposition, apprentissage non-supervisé, etc), utiliser une fonction déterministe permet de gagner en précision dès lors que la fonction utilisée est bien adaptée au problème traité. Ces fonctions peuvent-être par exemple un *min*, un *max*, un *ou*, ou bien d'autres fonctions plus complexes. Les figures 1.2 et 1.3 montrent respectivement un exemple de table de probabilité conditionnelle définis à partir des fonctions déterministes *ou* et *max*.

Cependant, la détermination des tables de probabilités à partir de fonctions déterministes présente l'inconvénient d'un manque de souplesse qui peut poser problème pour des cas particulier, des valeurs aberrantes.



$P(Y|X_1, X_2, X_3)$

X_1	X_2	X_3	0	1	2
0	0	0	1	0	0
0	0	1	0	1	0
0	0	2	0	0	1
0	1	0	0	1	0
...					
2	2	2	0	0	1

FIGURE 1.3 – Fonction déterministe : exemple du **max** (généralisation du **ou logique**)

Fonctions d'interactions canoniques

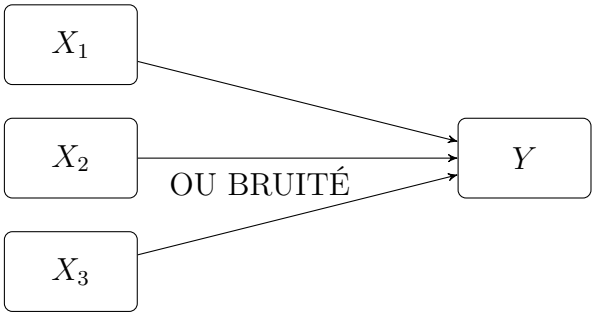
Pearl [102] décrit les interactions canoniques comme étant le lien entre une variable et l'ensemble de ses causes. Une interaction canonique y est dite disjonctive si chaque membre d'un ensemble de conditions est vraisemblable de causer un certain événement, et que l'occurrence de plusieurs de ces conditions ne diminue pas la vraisemblance de l'événement en question. Autrement dit, les causes sont chacune supposées être suffisantes pour causer l'effet, et leur capacité à causer l'effet est supposée indépendante de la présence d'autres causes.

Une alternative dite d'interaction conjonctive est également évoquée par Pearl [102], où les causes seraient non indépendantes et où la présence d'une des causes pourraient entraîner l'effet seulement si elle est associée à une autre, etc.

La porte ou bruité (*noisy or*) est une interaction canonique disjonctive, largement utilisée dans les réseaux bayésiens [100]. Les variables causes sont booléennes et sont notées X_1 ; X_2 ; ... ; X_n et la variable d'effet Y. De plus :

- chacune des causes X_i a une probabilité p_i d'être suffisante pour produire l'effet en l'absence de toutes les autres causes ;
- l'aptitude de chaque cause à être suffisante est indépendante de la présence d'autres causes.

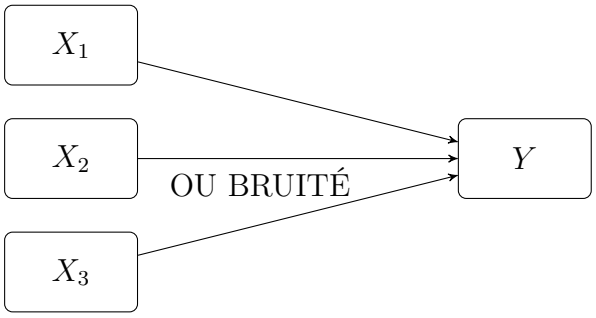
Les deux hypothèses ci-dessus nous permettent de spécifier la totalité de la distribution de probabilité conditionnelle avec seulement n paramètres p_1 ; p_2 ; . . . ; p_n. p_i représente la probabilité que l'effet Y soit vrai si la cause X_i est présente et que toutes les autres causes X_j , $j \neq i$, sont absentes. La figure 1.4 représente la forme générique de la porte ou bruité avec les p_i. La figure 1.5 montre un exemple de ou bruité avec $p_1 = 0.7$, $p_2 = 0.8$,



$P(Y|X_1, X_2, X_3)$

X_1	X_2	X_3	F	T
F	F	F	1	0
F	F	T	$1-p_3$	p_3
F	T	F	$1-p_2$	p_2
F	T	T	$(1-p_2)\times$ $(1-p_3)$	$1-((1-p_2)\times$ $(1-p_3))$
...				
T	F	F	$1-p_1$	p_1
T	T	T	$(1-p_1)\times$ $(1-p_2)\times$ $(1-p_3)$	$1-((1-p_1)\times$ $(1-p_2)\times$ $(1-p_3))$

FIGURE 1.4 – Modèle canonique disjonctif : cas de la porte **ou bruité**



$P(Y|X_1, X_2, X_3)$

X_1	X_2	X_3	F	T
F	F	F	1	0
F	F	T	0.1	0.9
F	T	F	0.2	0.8
F	T	T	0.02	0.98
...				
T	T	T	0.006	0.994

FIGURE 1.5 – Exemple de porte **noisy or** avec $p_1 = 0.7$, $p_2 = 0.8$, $p_3 = 0.9$

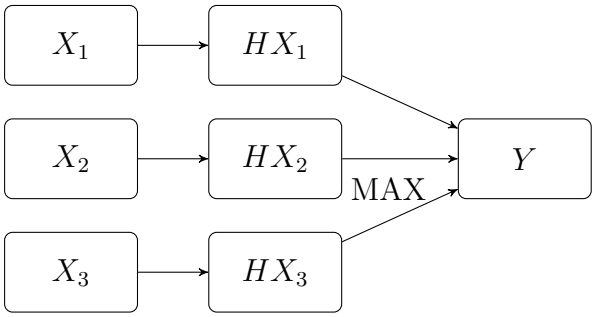
$p_3 = 0.9$.

$$p_i = P(y|\bar{x}_1, \bar{x}_2, \bar{x}_3, x_i, ..., \bar{x}_{n-1}, \bar{x}_n) \tag{1.7}$$

Onisko et al [100] détaillent que les formules de l'équation 1.7 et 1.8 découlent des affirmations précédentes, avec $\mathbf{X}_p$ l'ensemble des X_i présents, c'est-à-dire que $X_i \in \mathbf{X}_p \iff x_i = T$. Enfin, celles-ci permettent de construire entièrement la table de distribution conditionnelle de Y.

$$P(y|\mathbf{X}_p) = 1 - \prod_{i:X_i \in \mathbf{X}_p} (1 - p_i) \tag{1.8}$$

Diez et Druzdzel [31] définissent une équivalence à la définition précédente donnée par Pearl de la porte ou bruité, qui comprend des paramètres bruités (*noisy parameters* : les



$P(HX_3|X_3)$

X_3	0	1	2
0	1	0	0
1	1-(p_{11}+p_{12})	p_{11}	p_{12}
2	1-(p_{21}+p_{22})	p_{21}	p_{22}

$P(Y|HX_1, HX_2, HX_3)$

X_1	X_2	X_3	0	1	2
0	0	0	1	0	0
0	0	1	0	1	0
0	0	2	0	0	1
0	1	0	0	1	0
...					
2	2	2	0	0	1

FIGURE 1.6 – Modèle canonique disjonctif : équivalence de la porte **noisy max** proposée par Diez et Druzdzel [31].

variables HX_i) suivi d'un *ou* classique. Ils appellent ces équivalences des *Noisy ICI models* et cette forme est présentée dans la figure 1.6 sous sa généralisation max bruité.

Le max bruité (*noisy max*) est une généralisation du ou bruité [30, 53]. Les sorties ne sont plus *vrai* (T) ou *faux* (F), mais n'importe quelle valeur du domaine de la variable d'effet. Par exemple, dans la figure 1.6, les variables peuvent prendre les valeurs 0, 1, ou 2. Le max bruité est souvent utilisé pour représenter des modèles causaux [117].

Zagorecki et Druzdzel [138] explorent des réseaux bayésiens construits par expertise pour y chercher des portes max bruité qui n'avaient pas été explicitées, et proposent un algorithme pour transformer une table de distribution conditionnelle en porte max bruité. Cette transformation peut se faire sous certaines conditions : 1) le nœud enfant et tous ses parents doivent être des variables pour lesquelles des valeurs élevées correspondent à une occurence de l'effet ; 2) chacun des nœuds parents doit représenter une cause qui peut produire l'effet (le nœud enfant) en l'absence des autres causes, et 3) il ne doit pas y avoir de corrélation significative entre les causes. Takikawa et d'Ambrosio [117] proposent une méthode de factorisation de max bruité pour réduire les temps de calculs lors de son utilisation.

1.3 Extension aux variables continues

Lorsqu'une variable à intégrer dans un réseau bayésien est définie sur un domaine continu, il est possible de discrétiser préalablement cette variable pour obtenir un modèle totalement discret, ou encore de conserver l'aspect continu de cette variable en l'intégrant telle quelle dans le modèle.

1.3.1 Discrétisation

Le processus de discrétisation consiste à prendre en entrée des valeurs numériques continues et à les associer à des intervalles de discrétisation. Aguilera et al [2] rendent compte du fait qu'environ 50% des travaux procédant à une discrétisation avec des réseaux bayésiens ne détaillent pas la méthode de discrétisation utilisée, ce qui complexifie l'étude de ce champ de travaux.

Le choix de la méthode de discrétisation peut faire considérablement varier les performances finales [99], de même que le choix du nombre d'intervalles de discrétisations [38]. Les valeurs continues peuvent être gérées de plusieurs façons : leur discrétisation peut être manuelle, supervisée ou non-supervisée [10].

Discrétisation manuelle

La méthode la plus fréquemment utilisée [3] est la discrétisation manuelle [106, 105]. Elle consiste à ce qu'une personne, souvent un expert du domaine d'application, détermine les seuils des intervalles de discrétisation, en se basant sur ses connaissances et éventuellement à l'aide d'outil pour la compréhension des données tels que des histogrammes [10].

L'utilisation de cette méthode permet donc d'incorporer des connaissances d'expert, les intervalles finaux sont interprétables et enfin il ne requiert pas forcément de calculs sur ordinateur [10]. L'inconvénient est que cette la discrétisation manuelle est difficilement applicable si un expert n'est pas disponible pour passer du temps à la mettre en place.

Discrétisation non-supervisée

La discrétisation non-supervisée est utilisée lorsqu'aucun expert n'est disponible pour déterminer la discrétisation. Les algorithmes *equal-width (EW)* et *equal-frequency (EF)* [63] sont fréquents dans ce cas. Le premier consiste à diviser le domaine de définition en

un nombre donné d'intervalles de tailles égales, ce qui fonctionne plutôt bien pour des variables aux distributions uniformes mais qui génère des distributions de probabilités dégénérées hors de ce cas idéal [10]. Le deuxième algorithme, *EF* définit des intervalles de fréquences égales, ce qui a tendance à dissimuler les valeurs extrêmes, qui sont souvent intéressants dans un cas applicatif. Le nombre d'intervalles utilisé en pratique est généralement compris entre 2 et 10 [124]. Sari et al [111] utilisent quant à eux une discrétisation préalable à l'aide de l'algorithme des k-moyennes.

Ces algorithmes ont l'avantage d'être simple à mettre en œuvre et de nécessiter peu de coûts de calculs, cependant une discrétisation préalable à l'apprentissage, basée uniquement sur la distribution de chaque variable indépendemment peut faire perdre beaucoup d'informations. De plus, les intervalles appris ne sont généralement pas interprétables.

La discrétisation non-supervisée présente l'intérêt d'être globalement peu gourmand en calculs, de ne pas nécessiter le connaissance *a priori* ni d'expert, et enfin d'être globalement résistant au bruit dans les données [10]. Cependant, les intervalles appris n'ont généralement pas de signification, et ils peuvent avoir peu d'intérêt au regard de la prédictivité du modèle car l'apprentissage ne prend pas en compte la variable cible.

Discrétisation supervisée

La discrétisation supervisée est la moins utilisée des techniques de discrétisation, elle consiste à utiliser la sortie finale de la classe pour optimiser la discrétisation de chaque variable d'entrée.

Dougherty et al [32] prouvent leur méthode de discrétisation supervisée utilisée avec un classifieur bayésien naïf (figure 1.7) comme étant meilleure que la discrétisation non-supervisée sur un large ensemble de jeux de données. Les méthodes basées sur l'entropie telles que celles proposées par Fayyad et al [36] ou Kononenko [72] sont fréquemment utilisées [10]. Cependant, la discrétisation supervisée pour les réseaux bayésiens reste la moins utilisée dans la littérature.

L'avantage de l'apprentissage de la discrétisation de façon supervisée est que ce dernier va optimiser la capacité prédictive du modèle et offre généralement de meilleurs résultats qu'un apprentissage non-supervisé. Cependant, de la même façon que pour l'apprentissage non-supervisé, les intervalles appris peuvent ne pas être interprétables.

Aucune méthode de discrétisation n'est reconnue comme étant globalement meilleure que les autres, et la méthode à utiliser va fortement dépendre de l'application. En effet, dans le cas où il faut respecter une signification pré-déterminée des valeurs, la discréti-

sation manuelle sera en général bien adaptée. Si le but est de tester beaucoup de jeux de données, il vaudra certainement mieux utiliser une méthode automatique, etc. Il est cependant fortement recommandé, si possible, de vérifier l'impact de plusieurs discrétisations différentes sur le résultat final d'un modèle appris à partir de données, ainsi que sur ses performances [99].

Pour éviter la perte d'information liée à un pré-traitement, il est possible d'apprendre la discrétisation de chaque variable en même temps que le reste du modèle.

1.3.2 Modèle hybride

Un réseau bayésien hybride est un modèle qui intègre des variables définies dans un domaine discret et d'autres dans un domaine continu.

Linear conditional gaussian bayesian network (LCGBN) est un terme général pour les réseaux bayésiens incluant des distributions gaussiennes conditionnelles dans la définition de certaines variables. La distribution d'une variable continue peut être exprimée à l'aide d'une ou plusieurs gaussiennes [93].

Song et al [114] proposent une autre méthode probabiliste, *Gaussian process latent variable model* (GP-LVM) basée sur du *clustering* pour la discrétisation multivariée de données à haute dimension et à forte corrélation. Cette approche offre de bons résultats en classification mais perd de l'information car elle n'exploite pas la structure du réseau.

La méthode de discrétisation multivariée [82] permet de maximiser l'information utilisée grâce à une méthode de discrétisation utilisant un mélange de gaussiennes. Pour éviter l'immense coût calculatoire d'apprendre conjointement la structure du réseau bayésien et la discrétisation des variables, cette méthode fonctionne par itération en alternant l'apprentissage du graphe avec une discrétisation fixée, et une optimisation de la discrétisation avec une structure donnée. De la même façon, l'étape de la discrétisation fixe les valeurs de toutes les variables sauf une, et optimise les gaussiennes de discrétisation des variables une à une jusqu'à ce qu'un certain critère soit rencontré.

1.4 Utilisation en classification et diagnostic

Les réseaux bayésiens sont des modèles polyvalents qui peuvent être utilisés pour plusieurs objectifs. La classification est un usage très fréquent des modèles prédictifs de manière générale et de nombreuses études ont été effectuées pour utiliser les réseaux

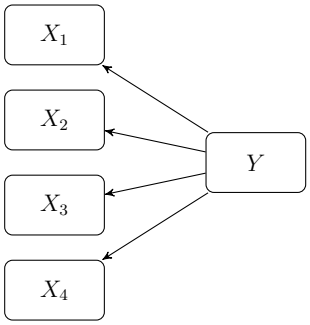


FIGURE 1.7 – Classifieur Bayésien naïf

bayésiens dans ce sens (partie 1.4.1). Les réseaux bayésiens présentés peuvent être utilisés pour divers domaines d'application (partie 1.4.3).

1.4.1 Réseau bayésien pour la classification

Le classifieur bayésien le plus simple et classique est le classifieur bayésien naïf (figure 1.7). C'est une structure simple, dans laquelle le nœud de la variable classe est le seul parent de chaque autre variable. La construction du classifieur bayésien naïf repose sur l'hypothèse que toutes ses variables sont indépendantes sachant la classe, ce qui s'exprime comme dans l'équation 1.9 [109].

$$P(\mathbf{X}|y_k) = \prod_{j=1}^{n} P(X_j|y_k) \tag{1.9}$$

Cette hypothèse d'indépendance entre les variables est en théorie peu réaliste, car les variables sont très rarement indépendantes dans un problème de classification. Cependant, le classifieur bayésien naïf offre en pratique de bonnes performances dans de nombreuses tâches de classification, surtout quand les différentes variables ne sont pas fortement corrélées [77]. Les bonnes performances de ce modèle simple en ont fait un axe de recherche privilégié et a donné lieu à de multiples études, donnant pour résultat tout un éventail d'améliorations possible. Bielza et al [12] proposent une hiérarchisation de ces modèles par ordre croissant de complexité (voir figure 1.8).

De nombreuses variantes structurelles du classifieur bayésien naïf ont également été proposées, entre autre le *tree augmented* bayésien naïf ou le *BN augmented* bayésien naïf. Le *tree augmented* bayésien naïf (TAN, voir figure 1.9) est construit de la même façon qu'un classifieur bayésien naïf classique, puis un arbre est appris sur l'ensemble des variables privé de la classe. Le *BN augmented* bayésien naïf (BAN, voir figure 1.10) quant

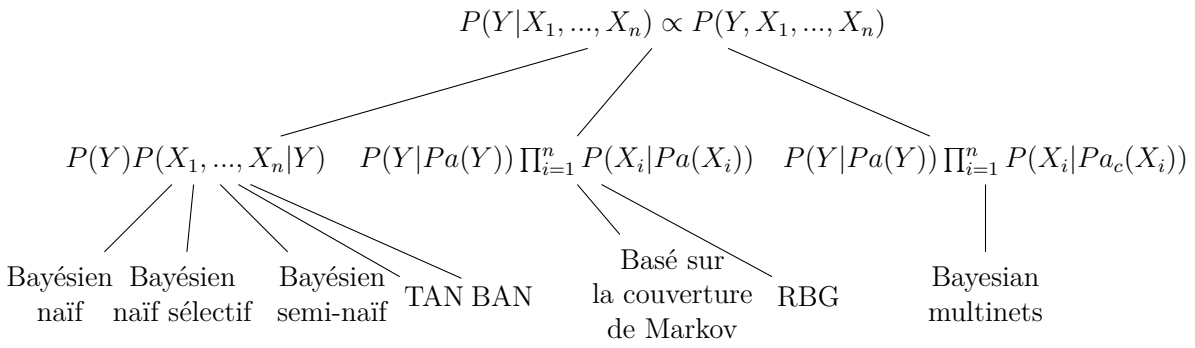


FIGURE 1.8 – Catégorisation des classifieurs de réseaux bayésiens discrets en fonction de la factorisation de $P(\mathbf{X}, Y)$. Illustration issue des travaux de Bielza et al [12].

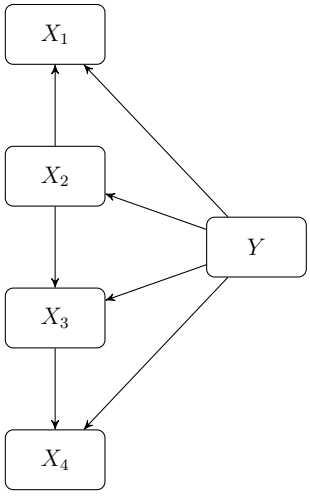


FIGURE 1.9 – Exemple de *tree augmented* Bayésien naïf

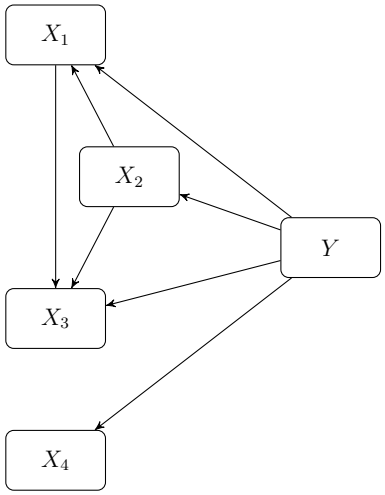


FIGURE 1.10 – Exemple de *BN augmented* Bayésien naïf (BAN)

à lui, enlève une contrainte par rapport au TAN car il consiste également à former un graphe entre l'ensemble des variables privé de la classe, mais n'impose pas que ce graphe soit un arbre.

Enfin, un classifieur peut également être construit à partir d'un réseau bayésien généraliste (RBG), qui est un réseau bayésien qui n'a pas de structure imposée et pour lequel le nœud de la variable classe est traité comme n'importe quel autre. Après une phase d'apprentissage de la structure, si les données sont complètes, les nœuds qui sont en dehors de la couverture de Markov peuvent être supprimés du nœud de classification. La couverture de Markov d'un nœud englobe ses parents, ses enfants et les autres parents de ses enfants.

Les différentes variantes du classifieur bayésien naïf ont plusieurs avantages. Le fait que l'espace de recherche de structure soit considérablement réduit permet un temps d'apprentissage plus rapide, surtout dans le cas du TAN, et leur donne l'avantage de pouvoir intégrer plus facilement des connaissances d'expert [18].

1.4.2 Sélection de variables

Lors de l'utilisation de modèles pour la classification automatique, la question du choix des variables utilisées en entrées se pose. Hoque et al [54] énumèrent quatre types d'approches différentes pour la sélection de variables. La première est l'approche par filtre, qui se fait hors du processus d'apprentissage du modèle [47] ; les variables peuvent être filtrées (*filter approach*) grâce à une mesure de corrélation, ou un *clustering* avec des

k-moyennes par exemple. L'avantage de cette méthode est son effacitée ; de très nombreuses variables peuvent être filtrées en très peu de temps. La deuxième approche, dite enveloppante (*wrapper approach*), utilise comme métrique la précision du modèle en tant que classifieur. Ce type de méthode donne généralement de bons résultats mais est plus demandeur en terme de puissance calculatoire [14, 65]. Les méthodes dite "intégrées" (*embedded approach*) reposent sur le même principe que les méthodes enveloppantes mais sont spécifiques à chaque algorithme [54, 47]. Enfin, il existe des approches mixtes (*hybrid approaches*), qui mêlent les approches enveloppantes et celles par filtre ; la méthode par filtre pré-sélectionne un ensemble de variables et ce dernier est ensuite affiné avec la méthode enveloppante [55]. Cette approche permet de conserver les avantages des deux méthodes.

Dans le cas de l'utilisation d'un modèle probabiliste tel qu'un réseau bayésien, il est possible de filtrer les variables à l'aide de l'information mutuelle, qui est la quantité d'information que l'on va obtenir sur une variable aléatoire lorsqu'une autre sera mesurée. Dans le cas discret, l'information mutuelle est définie par l'équation 1.10.

$$IM(X,Y) = \sum_{X_i \in x,y} P(X=x, Y=y) \log \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)} \tag{1.10}$$

La première proposition d'utilisation de l'information mutuelle pour la sélection de variables pour la classification a été faite par Battiti [8] avec l'algorithme MIFS *(mutual information feature selection)*. MIFS consiste à ajouter une à une les variables à l'ensemble de variables sélectionnées (initialisé vide) $\mathbf{S}$, en choisissant à chaque itération la variable qui maximise l'information mutuelle avec la variable classe en prenant en compte l'information mutuelle des autres variables choisies tel que dans l'équation 1.11, où β est un hyper-paramètre à choisir. Cette méthode est une approche par filtre. La procédure utilisée est une *forward selection*, c'est-à-dire une sélection par ajout de variable.

$$IM(X,Y) - \sum_{X_i \in \mathbf{S}} \beta \times IM(X, X_i) \tag{1.11}$$

Hoque et al [54] utilisent le même principe avec l'algorithme MIFS-ND mais en minimisant la moyenne des informations mutuelles entre variables là où Battiti [8] prend seulement en compte les variables déjà ajoutées à $\mathbf{S}$.

Ces approches basées sur des calculs d'information mutuelle sont des algorithmes itératifs, qui sont décrits comme étant exécutés jusqu'à obtenir un sous-ensemble de k variables. Cependant, il est parfois préférable de ne pas choisir un nombre de variables donné à l'avance, et de déterminer le sous-ensemble en se basant sur l'évolution d'un cri-

tère décrivant le modèle, tel que, par exemple, l'information mutuelle. Plusieurs méthodes proposent des solutions [139, 122, 112] pour trouver un coude dans un vecteur de données à deux dimensions, ce qui peut être appliqué dans le cas de l'évolution d'une métrique au cours du temps. La recherche d'un coude permet de sélectionner le moment où il est préférable d'arrêter l'optimisation car l'amélioration de la métrique sélectionnée a diminué [112]. Certaines des approches sont basées sur des mesures d'angle [139], ou des variations de moyennes dynamiques dans le cas de détection de coude au fur et à mesure de l'arrivée des données [110]. La méthode proposée par Satopää et al [112] sélectionne le coude dans les données en traçant une ligne entre les points (x_{min}, y_{min}) et (x_{max}, y_{max}). La figure 1.11 montre la façon dont sont exploitées les données : le point le plus éloigné de la ligne reliant (x_{min}, y_{min}) et (x_{max}, y_{max}) sur l'image (a) est le maximum local de la courbe de l'image (c). L'image (b) montre l'étape intermédiaire de calcul des valeurs de y pour la courbe résultante.

1.4.3 Domaines d'application

Les réseaux bayésiens offrent une généralité et une gestion de l'incertain qui est utile dans de nombreux domaines, notamment dans des cas critiques tels que des applications industrielles ou médicales, mais également dans des problèmes particulièrement complexes de par le nombre de variables à prendre en compte, telles que des problématiques environnementales.

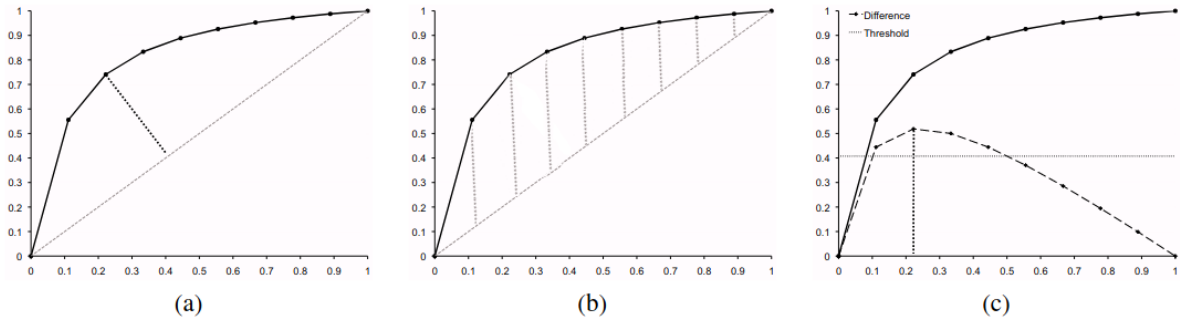


FIGURE 1.11 – Illustration du fonctionnement de Kneedle, algorithme de détection de coude, par Satopää et al [112]. (a) représente les données lissées et normalisées, le trait pointillé indiquant la distance perpendiculaire à y = x à la distance maximale. (b) montre les mêmes données, mais avec le trait pointillé de façon à être perpendiculaire à l'axe des abscisses. L'amplitude de ces barres sert ensuite à déterminer les valeurs des ordonnées des points de la courbe pointillée sur (c). Le coude est trouvé à x = 0,22.

Industrie

Hanea et al [50] proposent différentes grandes catégories d'application des réseaux bayésiens dans le domaine industriel :

— analyse de risque tels que solidité des barrages, transport aérien, etc ;
— fiabilité des structures, par exemple structure routière ;
— propriétés des matériaux, par exemple en apprendre plus sur les propriétés de l'asphalte pour améliorer sa qualité et augmenter sa durée de vie.

Environnement et gestion de ressources

La formalisation de modèles environnementaux est intrinsèquement complexe, en partie car l'information disponible est souvent très partielle.

Barton et al [7] traitent de problèmes de gestion de ressources, tel que la gestion du stock de poisson pour la pêche [125], ainsi que d'applications environnementales. Par exemple, Johnson et al [66] évalue la durabilité des populations de guépards en Namibie, Farmani et al [35] étudient la pollution nappes phréatiques. Hanea et al [50] se penchent également sur des problématiques de pollution en étudiant la corrélation entre la production de dioxyde de soufre et la concentration de particules fines sur toute la partie Est des Etats-Unis.

Médical

Langarizadeh et Moghbeli [76] proposent une vue d'ensemble sur l'utilisation d'un classifieur bayésien naïf pour le diagnostic de maladies telles que tumeur cérébrale, diabète, cancers, etc. Les résultats sont très bons (précision et spécificité supérieures à 80%), surtout pour des cas de métastase au cerveau dans le cas du cancer du poumon [131], de traumatisme crânien [71], de démence dans la maladie de Parkinson [90] ou encore de maladie coronarienne [17].

La figure 1.12 résume les domaines d'utilisation des réseaux bayésiens en médecine avec les proportions associées. Les réseaux bayésiens restent en proportion peu utilisés en pratique dans des applications médicales, milieu dans lequel des méthodes basées sur des régressions leur sont globalement préférés [4].

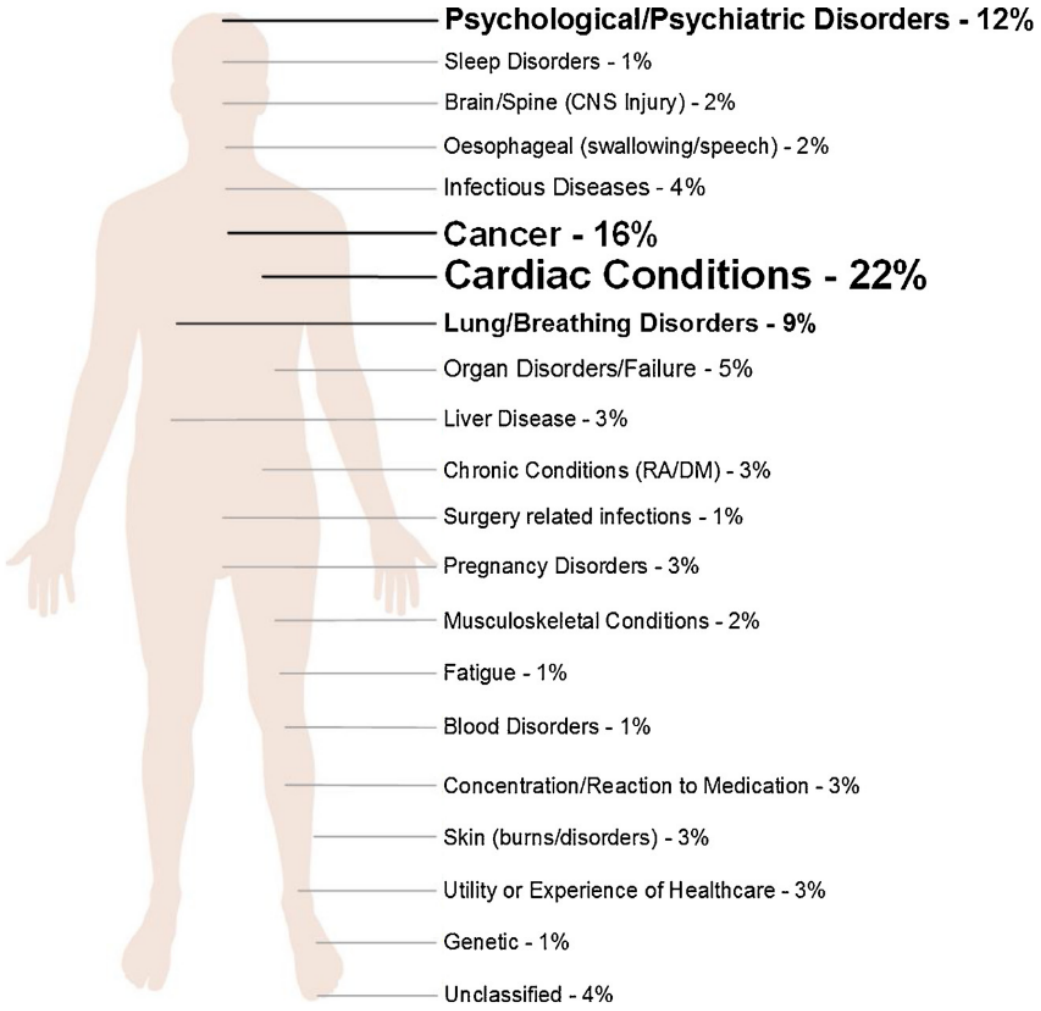


FIGURE 1.12 – Distribution des maladies étudiées dans la littérature sur les réseaux bayésiens. Illustration issue des travaux de McLachlan et al [85].

1.4.4 Diagnostic en usinage

Le diagnostic en usinage consiste à utiliser un classifieur pour prédire la présence de défaillance dans le système utilisé ou de défaut sur les pièces produites. Les problèmes en usinage sont souvent dûs à du broutement, à des vibrations, à l'usure de l'outil ou à l'endommagement de la machine-outil, et notamment de la broche.

Défaillance du système de production

Venkatasubramanian et al [127] répertorient trois types de dysfonctionnement différents :

— les erreurs de paramètres, en raison de variations de l'environnement qui ne sont pas inclues dans le modèle
— les changements structurels du processus, à cause de défaillance majeure de matériel
— les dysfonctionnements de capteurs

Ces travaux évoquent également un ensemble de critères d'évaluation d'un système de diagnostic, parmi lesquels l'explicabilité, la robustesse, le taux d'erreur, la rapidité de diagnostic.

Dey et al [29] utilisent des réseaux bayésiens dans le but de développer une méthodologie pour diagnostiquer la cause profonde lorsque le processus d'usinage ne se déroule pas comme prévu. Nguyen [95] construit également un modèle permettant de trouver la cause d'une défaillance dans un système de production à plusieurs étapes se déroulant dans plusieurs lieux différents.

Wang et al [132] utilisent un classifieur bayésien naïf, un TAN et RBG pour prédire les erreurs pendant l'usinage.

Qualité des pièces

Les problèmes de qualité en usinage sont souvent dues à l'usure de l'outil ou la rigidité de la pièce [95]. Correa et al [23] comparent l'efficacité d'un réseau bayésien et d'un réseau de neurones artificiel (RNA) pour la prédiction de la rugosité de la surface d'une pièce en usinage à grande vitesse (UGV), et concluent que les RB sont meilleurs en précision de classification, en temps de calcul, ainsi qu'en facilité d'interprétation.

Li et al [80] étudient la probabilité de la déformation de la pièce finale en fonction des conditions d'usinage, en prenant pour exemple l'incertitude des contraintes résiduelles respectivement initiales et celles induites par l'usinage.

1.5 Conclusion

Ce chapitre présente l'intérêt des réseaux bayésiens, notamment leur robustesse, ces modèles graphiques probabilistes existant depuis plusieurs dizaines d'années et ayant fait leurs preuves dans de nombreux cas théoriques et applicatifs. Ils ont la particularité d'être

très maniables. En effet, il est possible d'apprendre indépendamment la structure et les paramètres d'un réseau bayésiens. De plus, cet apprentissage peut se faire avec ou sans données, et ces données peuvent ne pas être complètes.

Enfin, comme de nombreux modèles probabilistes, ils ont l'avantage d'être très interprétables, ce qui rend leur utilisation très utile dans des domaines complexes et où chaque problème non détecté peut-être critique, dans lesquel les utilisateurs ont particulièrement besoin de comprendre les décisions qui sont prises par les modèles, tels que les domaines médicaux ou industriels. Des connaissances d'experts peuvent être ajoutées à tous les niveaux, de la discrétisation à la structure en passant par les distributions de probabilités, ce qui renforce leur adaptation dans les domaines où il peut être difficile d'obtenir des données, ou bien quand certaines valeurs ont une signification bien spécifique.

Notamment, le fait que les décisions des réseaux bayésiens soient explicables les rend particulièrement adaptés pour la classification et le diagnostic. C'est pourquoi de nombreuses variantes de réseaux bayésiens avec des contraintes de structures ont été proposées pour être encore plus facilement utilisable dans ce cadre. Parmi eux, le bayésien naïf est très utilisé, apprécié pour sa simplicité.

Un inconvénient des réseaux bayésiens, notamment dans le cadre du diagnostic, est de ne pas modéliser les variables sous une dimension temporelle. En effet, il peut être utile lorsque l'on veut évaluer l'usure à un moment donné, de prendre en compte certaines variables à un temps $t-1$, $t-2$, etc. C'est un pas nécessaire vers la maintenance prédictive. Il existe une extension temporelle des réseaux bayésiens, non présentées dans ce chapitre car ils sortent du contexte de notre étude : les réseaux bayésiens dynamiques.

En résumé

- ✓ De nombreuses méthodes sont disponibles pour l'apprentissage des réseaux bayésiens, à tous les niveaux de traitement, de manière supervisée ou non ;
- ✓ L'apprentissage de la structure et des paramètres peut se faire avec ou sans données ;
- ✓ Les variables continues peuvent être discrétisées en amont ou directement intégrées dans le modèle ;
- ✓ L'interprétabilité des RB en fait des outils de poids pour tous les domaines liés au diagnostic, notamment les domaines médicaux et industriels.

Chapitre 2

Co-training

Un nombre croissant de travaux porte sur l'étude du *co-training* et prouvent que ce dernier permet souvent d'améliorer les résultats en comparaison avec un apprentissage simple, lorsque les données sont naturellement scindées en plusieurs vues, ou dans bien des cas même si elles sont séparées artificiellement.

2.1 Définition

Le *co-training* consiste à apprendre conjointement deux modèles différents, généralement des classifieurs. Ce type d'apprentissage n'a pas de traduction française faisant consensus et sera donc nommé co-training de son nom anglophone pour le reste de ce manuscrit. La section 2.1.1 en donne une définition formelle, et la section 2.1.2 décrit les hypothèses théoriques qui garantissent quand elles sont respectées le bon fonctionnement du co-training, suivies dans la section 2.1.3 d'indicateurs qui permettent de d'évaluer le respect des hypothèses.

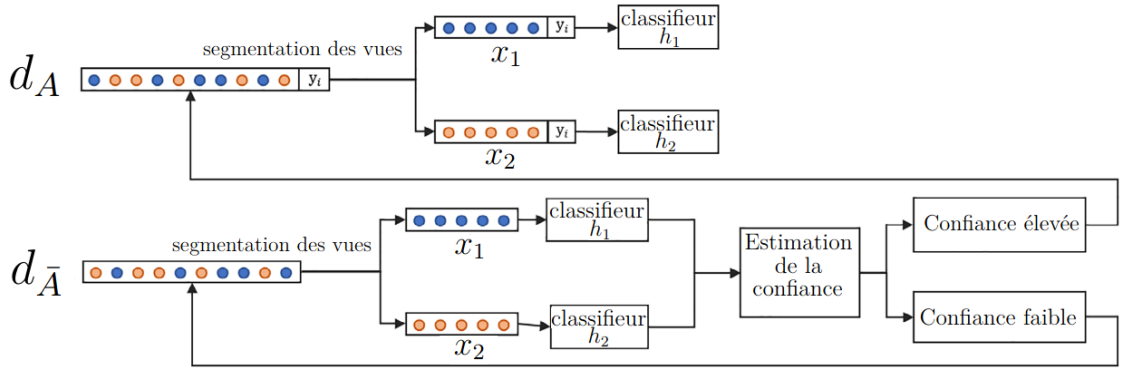


FIGURE 2.1 – Schéma générique co-training semi-supervisé. Illustration issue des travaux de Ning et al [97].

2.1.1 Définition

Les techniques d'apprentissage automatique peuvent se diviser en trois grandes catégories :

- l'apprentissage non supervisé, pour lequel l'ensemble du jeu de données utilisé n'est pas étiqueté ;
- l'apprentissage supervisé, pour lequel l'ensemble du jeu de données utilisé est étiqueté ;
- l'apprentissage semi-supervisé, qui se trouve à la croisée entre les deux précédents et qui utilise un jeu de données partiellement étiqueté.

Le co-training est classiquement un algorithme de classification semi-supervisé. Soit l'espace des instances $\mathcal{D} = \{\mathcal{X}^1, \mathcal{X}^2\}$, avec $\mathcal{X}^1$ et $\mathcal{X}^2$ les deux vues du jeu de données, c'est-à-dire deux ensembles disjoints de variables qui décrivent les mêmes objets à classer. De plus, $\mathcal{D}$ est divisé en deux parties $\mathcal{D}_A = \{\mathcal{X}^1_A, \mathcal{X}^2_A\}$ et $\mathcal{D}_{\bar{A}} = \{\mathcal{X}^1_{\bar{A}}, \mathcal{X}^2_{\bar{A}}\}$, qui sont respectivement l'ensemble des données annotées et non annotées.

Le principe est d'entraîner tout d'abord deux classifieurs h_1 et h_2 de manière supervisée, avec la partie du jeu de données étiquetée, respectivement sur $\mathcal{X}^1_A$ et $\mathcal{X}^2_A$. Par la suite, chaque classifieur étiquette certaines des instances initialement non étiquetés pour l'autre classifieur, et cela peut se faire itérativement, augmentant ainsi progressivement l'ensemble des données étiquetées. L'algorithme 2 détaille l'algorithme de la première proposition de co-training, par Blum et Mitchell [13]. La figure 2.1, une illustration issue des travaux de Ning et al [97] reprend cet algorithme et schématise un co-training semi-supervisé avec deux classifieurs pour deux vues. Cette figure présente le scindement qui est effectué entre

Algorithme 2 : Première proposition de co-training par Blum et Mitchell [13]

Entrée : $\mathcal{D}^{\mathcal{A}}$, $\mathcal{D}^{\bar{\mathcal{A}}}$, u, p, q

$\mathcal{D}^{\bar{\mathcal{A}}'} = u$ instances choisies aléatoirement dans $\mathcal{D}^{\bar{\mathcal{A}}}$

$\mathcal{D}^{\mathcal{C}} = \mathcal{D}^{\mathcal{A}}$

pour *i allant de 0 à k* **faire**

`// apprentissage des classifieurs`

h_1 = classifieur appris avec la vue 1 $\mathcal{X}_1^C$

h_2 = classifieur appris avec la vue 2 $\mathcal{X}_2^C$

`// étiquetage des données`

$l_p^{h_1} = p$ instances positives étiquetées avec h_1 dans $\mathcal{D}^{\bar{\mathcal{A}}'}$

$l_q^{h_1} = q$ instances négatives étiquetées avec h_1 dans $\mathcal{D}^{\bar{\mathcal{A}}'}$

$l_p^{h_2} = p$ instances positives étiquetées avec h_2 dans $\mathcal{D}^{\bar{\mathcal{A}}'}$

$l_q^{h_2} = q$ instances négatives étiquetées avec h_2 dans $\mathcal{D}^{\bar{\mathcal{A}}'}$

`// renforcement de l'ensemble de données étiquetées`

$\mathcal{D}^{\mathcal{C}} = \mathcal{D}^{\mathcal{C}} \bigcup \{l_p^{h_1}, l_q^{h_1}, l_p^{h_2}, l_q^{h_2}\}$

$\mathcal{D}^{\bar{\mathcal{A}}'} = 2p + 2q$ instances choisies aléatoirement dans $\mathcal{D}^{\bar{\mathcal{A}}}$

données annotées d_A et données non-annotées $d_{\bar{A}}$, et montre que les données non-annotées classées avec le plus de confiance deviennent des données annotées.

Dasgupta et al [26] ont donné la preuve de l'intuition fournie par la proposition de Blum et Mitchell [13] (voir algorithme 2).

Deux problèmes peuvent être soulevés concernant le co-training. Tout d'abord, la plupart des jeux de données, ne sont, dans les faits, pas composés de plusieurs vues. Ensuite, la précision des classifieurs étant faible au début de l'apprentissage, il est aisé que cela induise une mauvaise annotation des étiquettes et cela risque de créer une boucle de rétroaction négative en détériorant de manière croissante la capacité du modèle à généraliser.

La section 2.2 évoque des solutions pour limiter la boucle de rétroaction négative citée ci-dessus, et la section 2.3 donne des moyens d'adapter des situations pour permettre l'utilisation du co-training lorsque ce n'est pas possible à l'origine.

2.1.2 Hypothèses théoriques

Lors de la définition du co-training par Blum et Mitchell [13], il a été défini qu'il est attendu d'un algorithme de co-training qu'il ait de bonnes performances lorsque certaines hypothèses sont réunies. Il a été prouvé théoriquement que lorsque les deux hypothèses

suivantes sont vérifiées, le co-training est garanti de fonctionner [6].

1. Hypothèse de suffisance : chaque vue est capable de produire une prédiction correcte de la classe ;
2. Hypothèse d'indépendance : les vues sont conditionnellement indépendantes.

Tous les travaux concernant le co-training évoquent ces deux hypothèses d'indépendance et de suffisance. Cependant, de nombreux travaux théoriques et expérimentaux discutent la façon d'exprimer ces contraintes : pour certains, l'hypothèse d'indépendance est respectée si les vues ne sont pas trop fortement corrélées [1, 6].

2.1.3 Outils de mesure

Ling et al [81] proposent de traduire les hypothèses ci-dessus en formules basées sur un seuil pour quantifier numériquement le respect des dites hypothèses.

Les ensembles de variables des deux vues sont respectivement $\mathbf{X}^1$ et $\mathbf{X}^2$. Un classifieur est appris avec chaque vue pour prédire la classe cible, $p_{\mathcal{X}^1}$ et $p_{\mathcal{X}^2}$ étant respectivement la précision du classifieur des vues $\mathcal{X}^1$ et $\mathcal{X}^2$. p est la précision du classifieur appris avec les deux vues ensemble.

Hypothèse de suffisance

L'hypothèse de suffisance est décrite comme étant le fait que chaque vue produit individuellement une prédiction satisfaisante, ce qui signifie que p, $p_{\mathcal{X}^1}$ et $p_{\mathcal{X}^2}$ ne sont pas trop éloignées de 1, ce qui peut-être quantifié par le nombre δ_1, comme indiqué dans l'équation 2.1.

$$p > 1 - \delta_1, p_{\mathcal{X}^1} > 1 - \delta_1, p_{\mathcal{X}^2} > 1 - \delta_1 \tag{2.1}$$

Cette équation peut être transformée pour exprimer δ_1 en fonction des p, $p_{\mathcal{X}^1}$ et $p_{\mathcal{X}^2}$ comme dans l'équation 2.2.

$$\delta_1 < 1 - p, \delta_1 < 1 - p_{\mathcal{X}^1}, \delta_1 < 1 - p_{\mathcal{X}^2} \tag{2.2}$$

Enfin, l'équation 2.3 montre la façon d'obtenir δ_1 pour respecter les trois inégalités. δ_1 est appelée le seuil de suffisance.

$$\delta_1 = min(1 - p, 1 - p_{\mathcal{X}^1}, 1 - p_{\mathcal{X}^2}) \tag{2.3}$$

Hypothèse d'indépendance

Pour vérifier empiriquement l'indépendance conditionnelle des vues sachant la classe, il est possible de construire un classifieur qui prédit un x_i avec Y, et ce pour chaque valeur de la classe. Si les deux vues sont indépendantes, le classifieur ne devrait pas avoir une précision très supérieure à p'_{x_i}, qui est la fréquence de la valeur de classe majoritaire. La même expérience est effectuée avec un classifieur qui prédit y_j à partir de X, avec p'_{y_j} la fréquence de la valeur de classe majoritaire.

Les p_{x_i} et p_{y_j} peuvent donc être exprimés avec δ_2, le seuil d'indépendance, comme dans les équations 2.4 et 2.5.

$$p_{x_i} < p'_{x_i} + \delta_2, \forall i \in [0, s] \tag{2.4}$$

$$p_{y_j} < p'_{y_j} + \delta_2, \forall j \in [0, t] \tag{2.5}$$

A partir des équations 2.4 et 2.5, il est possible de déduire les équations 2.6 et 2.7.

$$\delta_2 > p'_{x_i} - p_{x_i}, \forall i \in [0, s] \tag{2.6}$$

$$\delta_2 > p'_{y_j} - p_{y_j}, \forall j \in [0, t] \tag{2.7}$$

Finalement, l'équation 2.8 donne la formule de calcul de δ_2 pour respecter les équations 2.6 et 2.7.

$$\delta_2 = max(max(p'_{x_i} - p_{x_i}), max(p'_{y_j} - p_{y_j})) \tag{2.8}$$

δ_1 et δ_2 sont par définition compris entre 0 et 1 et une valeur élevée signifie un respect moindre de l'hypothèse associée. Ling et al [81] concluent expérimentalement grâce à une méta-analyse que le co-training apporte globalement de meilleures performances qu'un entraînement simple lorsque $\delta_1 < 0,23$ et $\delta_2 < 0,15$, et que l'utilisation de co-training aurait plutôt tendance à dégrader les résultats si $\delta_1 > 0,28$ et $\delta_2 > 0,26$.

2.2 Travaux existants sur le co-training

Le co-training a été défini à l'origine comme fonctionnant avec deux vues différentes et ayant pour objectifs l'apprentissage de deux classifieurs qui se transmettent des informations, le tout sous réserve du respect de certaines contraintes. Les contraintes nécessaires pour le fonctionnement du co-training sont largement discutées. De nombreux travaux évoquent également la manière de choisir les informations à passer d'un classifieur à l'autre, ainsi que d'autres façon d'exploiter la proposition originale du co-training.

2.2.1 Discussion sur les contraintes

Blum et Mitchell [13] prouvent empiriquement que le co-training peut fonctionner si l'hypothèse de suffisance et l'hypothèse d'indépendance conditionnellement à la classe sont respectées. Cependant, de nombreux travaux suivirent pour prouver qu'une condition moins contraignante au niveau de l'indépendance des vues permet d'atteindre le même résultat. Il existe également des travaux pour discuter le cas particulier de non suffisance des vues et les restrictions liées aux proportions des données étiquetées en entrées de l'algorithme.

Hypothèse d'indépendance

L'indépendance de chaque vue conditionnellement à la classe peut être exprimée par l'équation 2.9.

$$\forall x_1, x_2 : P(X_1 = x_1 | Y = y, X_2 = x_2) - P(X_1 = x_1 | X_2 = x_2) = 0 \tag{2.9}$$

Abney [1] décrit l'hypothèse d'indépendance conditionnellement à la classe comme "non-raisonnable" car étant une hypothèse très forte, et propose une alternative plus souple, qui correspond à la formule dans l'équation 2.10, avec un $\epsilon \geq 0$. Cette situation est énoncée comme étant plus réaliste.

$$\forall x_1, x_2 : |P(X_1 = x_1 | Y = y, X_2 = x_2) - P(X_1 = x_1 | X_2 = x_2)| < \epsilon \tag{2.10}$$

Enfin, Balcan et al [6] se basent sur l'hypothèse que chaque classifieur n'a jamais tort s'il est très confiant, et sa proposition d'évaluation de "l'indépendance relaxée" repose autant sur les données en elles-mêmes que sur les sorties des classifieurs. La proposition d'α-expansion se retrouve en opposition aux deux hypothèses précédemment décrites qui

se détachaient entièrement des modèles utilisés, et elle se positionne également comme étant moins contraignante.

La proposition d'α-expansion formule le fait que le co-training ne sera utile que si les vues ont des informations complémentaires à apporter. Par conséquent, dans le cadre des prédictions positives de la classe, si les deux classifieurs sont toujours confiants sur les mêmes instances des données à classifier, le co-training n'a pas d'intérêt. Dans cette optique, la définition tient compte de la notion de confiance dans la prédiction pour déterminer si le co-training va améliorer les performances. c_i correspond à la confiance du $i$ème classifieur : par exemple c_1^+, c_1^- sont respectivement à interpréter comme le classifieur 1 est confiant ou non dans sa prédiction.

$$P(x_1, x_2|Y = y, c_1 \neq c_2) \geq \alpha min(P(x_1, x_2|Y = y, c_1^+, c_2^+), P(x_1, x_2|Y = y, c_1^-, c_2^-) \quad (2.11)$$

Si l'équation 2.11 est vraie $\forall x_1, x_2$, cela signifie que les deux classifieurs ne sont pas confiants dans les mêmes prédictions dans la plupart des cas, et donc que le co-training devrait être bénéfique pour l'apprentissage.

Hypothèse de suffisance

Wang et Zhou [133] discutent le cas dans lequel l'hypothèse de suffisance ne serait pas respectée. Les conclusions de ces travaux sont que les modèles peuvent être affectés par deux tendances lorsque les vues ne sont pas considérées comme suffisantes : le bruit d'étiquetage (*label noise*), qui correspond à un mauvais étiquetage de certaines instances, et le biais d'échantillonage (*sampling bias*), qui est causé par le fait d'essayer de réduire le bruit d'étiquetage (par exemple en n'ajoutant au jeu de données que les instances dont la classification est des plus certaines).

Proportion dans les données étiquetées

Pour une utilisation de co-training en semi-supervisé comme c'est souvent le cas, une des questions venant à l'esprit est la proportion de données étiquetées nécessaire. Matsubara et al [84] effectuent une étude empririque sur l'impact de la proportion de la classe cible des données étiquetées par rapport aux données globales. Le classifieur utilisé est un classifieur bayésien naïf et la conclusion des travaux est que les meilleures performances sont généralement atteintes lorsque la proportion de la classe cible dans la réalité est égale

à la proportion dans les données annotées.

Il est rare que la question de la proportion des classes soit évoquée dans les articles de recherche, et il serait intéressant d'avoir plus de visibilité à ce sujet, comme le soulignent Ning et al [97].

2.2.2 Choix des étiquettes

Le processus de co-training comprend une phase d'ajout des étiquettes à des instances non étiquetées. Une partie des recherches sur le co-training s'intéresse à savoir quelles étiquettes ajouter à chaque étape. Plusieurs méthodes ont été explorées, notamment dans le domaine de l'apprentissage actif, ou du tri-training avec des variantes impliquant non pas deux mais trois classifieurs, ou encore d'autres techniques déviant du fonctionnement classique du co-training.

Apprentissage actif

Plusieurs sortes de variantes au co-training ont été proposées dans le domaine de l'apprentissage actif, qui consiste à ce que l'humain intervienne pour orienter l'algorithme pendant l'apprentissage. Le co-training corrigé proposé par Pierce et al [104] est similaire au co-training classique, mais requiert de l'homme qu'il corrige certaines annotations effectuées par le modèle. Cette variante, est un compromis entre la perte de temps engendrée par le fait d'annoter, et l'augmentation de la précision sur les prédictions. Cet algorithme est tout de même un gain de temps pour l'annotateur par rapport au fait d'annoter plus de données dans le jeu de données d'entrée, il n'aura ici qu'à corriger les prédictions erronées. Sur la base du co-training corrigé, Hwa et al [57] introduisent le co-training corrigé unilatéral, qui consiste à ne corriger les sorties que d'un seul des deux classifieurs, pour un temps passé à annoter divisé par deux pour l'humain, et une faible baisse de précision.

Muslea et al [94] présentent une famille d'algorithme nommée co-testing, appartenant à la catégorie des algorithmes de *selective sampling*. Le principe général est de sélectionner certaines des instances sur lesquels les classifieurs ne sont pas en accord, et de les annoter à la main. Une proposition pour la base des expérimentations est l'algorithme appelé co-testing naïf, et consiste à sélectionner aléatoirement une partie des exemples sur lequel les différents classifieurs sont en désaccord.

Deux variantes au co-training classique ont notamment été proposées : la version basée sur l'accord [48] et la version basée sur le désaccord. Clark et al [19] proposent

un algorithme de co-training basé sur l'accord entre les classifieurs. Guz et al [48] se placent dans le cadre de l'hypothèse dite de faible dépendance entre les vues, et utilise un algorithme de co-training semi-supervisé basé sur l'accord entre les classifieurs sur les données à étiqueter. Cet algorithme surpasse le co-training classique sur une application de segmentation de phrases, pour la classification de noms propres. Une propriété y est avancée : le taux de désaccord entre les classifieurs correspond à une borne supérieure sur le taux d'erreur du co-training.

Tri-training

Le tri-training est une extension du co-training, se basant sur le même principe mais utilisant trois modèles au lieu de deux pour l'apprentissage. Zhou et al [142] se basent sur un principe de vote entre les classifieurs, c'est-à-dire qu'une étiquette est ajouté à l'ensemble de données annotées si les deux autres classifieurs sont d'accord concernant cette étiquette, et si certaines autres conditions sont respectées (basées sur des estimations des taux d'erreur de classification des classifieurs). Bhalgat et al [11] utilisent un réseau Bayésien naïf, un réseau neuronal (perceptrons à plusieurs couches) et une machine à support de vecteurs dans un paradigme d'apprentissage avec deux professeurs et un élève en tri-training.

Autres

Zhou et Li ont proposé un algorithme de co-forêt [79], un apprentissage en co-training de forêts aléatoires pour savoir comment sélectionner les données non étiquetées avec la plus grande confiance. En effet, des études expérimentales [87] ont montré que l'élagage peut réduire l'impact du bruit dans les données en évinçant les parties faiblement informatives.

Parmi les méthodes citées ci-dessus, le mécanisme de vote présente l'inconvénient que les classifieurs peuvent être d'accord mais peu certains, auquel cas il y a plus de risque de mal annoter les données. Cela peut-être éventuellement corrigé grâce à la correction manuelle par les humains (apprentissage actif), ou par le fait de prendre en compte la confiance dans l'étiquetage au moment de prendre la décision.

2.2.3 Détournement de l'algorithme original

Le co-training est généralement une utilisation de deux modèles, classiquement similaires, et utilisant deux vues pour la classification. Certaines études portent sur l'usage du co-training avec différents types de classifieurs, en utilisant parfois une seule vue, ou encore en faisant de la régression par exemple.

Plusieurs classifieurs différents

Kiritchenko et al [70] appliquent le co-training pour la classification d'e-mails, en créant une vue pour le sujet du mail et une vue pour le corps du mail. Cette étude obtient de meilleures performances sur le co-training semi-supervisé avec les machines à supports de vecteurs que le classifieur bayésien naïf. Kiritchenko et al [70] expliquent que les mauvaises performances du classifieur bayésien naïf sont probablement dues au fait que le jeu de données ait de nombreuses variables et en particulier que la condition d'indépendance ne soit pas respectée. Cette explication semble confirmée par le fait que les résultats sont bien meilleurs après une sélection de variables.

Certains travaux mélangent différents types de classifieurs, comme Ju et al [67] avec

Algorithme 3 : Variante du co-training basée sur le désaccord

Entrée : ensemble L et de données respectivement étiquetées et non-étiquetées

1 U' = u instances choisies aléatoirement dans U
2 **pour** *i allant de 0 à k* **faire**
  `// apprentissage des classifieurs`
3 h_1 = classifieur appris avec la partie x_1 de x dans L
4 h_2 = classifieur appris avec la partie x_2 de x dans L
  `// étiquetage des données`
5 $l_{p,h_1} = p$ instances positives étiquetées avec certitude par h_1 et incertitude par h_2 dans U'
6 $l_{n,h_1} = n$ instances négatives étiquetées avec certitude par h_1 et incertitude par h_2 dans U'
7 $l_{p,h_2} = p$ instances positives étiquetées avec certitude par h_2 et incertitude par h_1 dans U'
8 $l_{n,h_2} = n$ instances négatives étiquetées avec certitude par h_2 et incertitude par h_1 dans U'
  `// renforcement de l'ensemble de données étiquetées`
9 L = L + $\{l_{p,h_1}, l_{n,h_1}, l_{p,h_2}, l_{n,h_2}\}$
10 U' = $2p + 2n$ instances choisies aléatoirement dans U

des réseaux de neurones convolutionnels et des réseaux de neurones récurrents.

Une seule vue

Comme mentionné précédemment, certains travaux dévient le co-training de son objectif original en n'utilisant non pas deux mais une seule vue. Goldman et al [42] mettent à profit deux algorithmes utilisés classiquement en apprentissage supervisé mais avec une seule vue, en utilisant l'un pour étiqueter les données de l'autre. Zhou et al [141] effectuent l'apprentissage également avec une seule vue apprise par deux variantes de paramètres différents sur le même type de classifieur.

Variantes à la classification

Le co-training a également été utilisé pour de la régression en semi-supervisé par Zhou et al [141], avec une méthode basée sur les k-plus proches voisins.

Collins et al [20] introduisent DL-CoTrain, basé sur la proposition originale de Blum et Mitchell [13] et Yarowski [136], ce dernier étant un algorithme non-supervisé pour l'analyse textuelle. DL-CoTrain sert à la reconnaissance et la classification spécifiquement pour les entitées nommées.

Des adaptations du co-training ont également été développées pour des domaines plus larges que la prédiction, tels que l'adaptation de modèle avec la *co-adaptation* proposée par Tur et al [123], testé sur une tâche de segmentation de dialogue. Niu et al [98] proposent une variante dans laquelle, au lieu d'utiliser le mécanisme du co-training d'étiquetage des données par les modèles, les deux classifieurs sont entraînés sur les deux vues en étant encouragés par une fonction de coût à produire les mêmes étiquettes.

2.3 Validation des algorithmes de co-training

Il existe des benchmarks disponibles pour la validation d'algorithmes de co-training, ceux-ci sont présentés dans la section 2.3.1. Lorsque les benchmarks n'existent pas pour un problème donné, il est également possible de créer un jeu de données à deux vues à partir d'un jeu de données à une vue, certaines des méthodes existantes sont répertoriées dans la section 2.3.2.

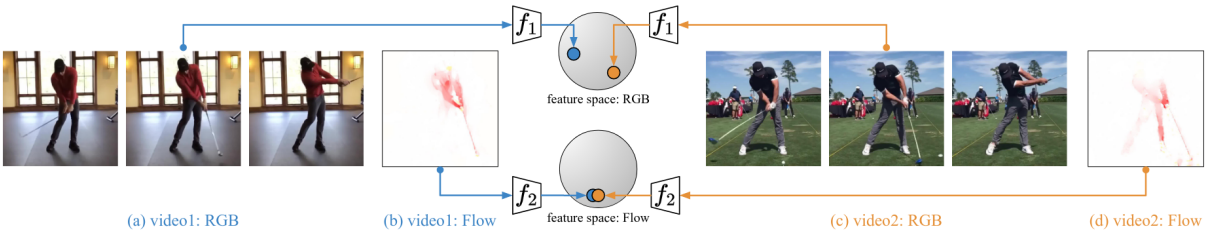


FIGURE 2.2 – Exemple de co-training avec le flux RGB et le flux optique dans une vidéo de golf. Illustration de Han et al [49]. Le flux RGB est très différent dans les deux exemples mais le flux optique est très semblable.

2.3.1 Benchmark existants

Les benchmarks disponibles pour le co-training sont par exemple des séquences génomiques, des bases de caractères pour la segmentation de mots chinois, ou encore des données d'imageries photos et vidéos.

Données vidéo

Han et al [49] proposent du co-training pour de l'analyse vidéo, plus exactement pour identifier des actions dans des vidéos. Leur jeu de données reprend les benchmarks **UCF101** [115] [1], **K400** [68] [2] et **HMDB51** [73] [3]. Une des vues est composée du flux RGB et l'autre du flux optique, qui est entre autres utilisé pour la détection du mouvement et la segmentation d'objets. La figure 2.2 illustre le flux RGB et le flux optique dans un exemple de vidéo de golf.

Analyse du génome

La base de données miRBase [43] [4] regroupe des séquences de micro-ARN de milliers d'espèces. Sheikh Hassani et al [113] utilisent cette base de données pour en faire un benchmark de co-training, en séparant en deux vues les variables liées respectivement à la séquence et à l'expression des gènes. Le benchmark de co-training résultant comprend six espèces animales (homme, souris, drosophile, vache, poulet, cheval). La cible est la classification des séquences en micro-ARN (positif si la séquence est effectivement du

1. `https://www.crcv.ucf.edu/data/UCF101.php`
2. `https://www.deepmind.com/publications/the-kinetics-human-action-video-dataset`
3. `https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/`
4. `https://mirbase.org/ftp.shtml`

micro-ARN), avec pour chaque espèce des dizaines de milliers d'entrées, dont quelques centaines d'instances étiquetées positives et négatives, vérifiées en laboratoire.

Traitement d'image et autre

Zhou et al [140] et Wang et al [134] utilisent un mélange entre du co-training semi-supervisé et du co-testing, respectivement pour de la recherche d'image et de classification de page web avec le benchmark utilisé par [13] [5] : 1051 pages web collectées sur les sites web des départements informatique de quatre universités : Cornell, Université de Washington, Université du Wisconsin, et Université du Texas. Pour les pages web, une vue est composée par tous les mots compris dans la page, et l'autre contient tous les mots des liens hypertexte des autres pages pointant vers celle concernée.

2.3.2 Création de benchmarks pour le co-training

La création d'un benchmark pour le co-training consiste à prendre un jeu de données classique $\mathcal{D}$ à une vue, et à le transformer en jeu de données à deux vues $\mathcal{X}^1$, $\mathcal{X}^2$.

Les travaux de Feger et al [37] se penchent sur la question de la séparation des vues à partir d'une approche par graphes, en calculant l'information mutuelle conditionnelle pour mesurer l'indépendance entre les vues. Les équations 2.12 et 2.13 sont respectivement les formules de l'information mutuelle et de l'information mutuelle conditionnelle.

$$\begin{aligned} IM(X,Y) &= H(X) - H(X|Y) \\ &= \sum_x P(x) \log_2 \frac{1}{P(x)} - \sum_x \sum_y P(x,y) \log_2 \frac{1}{P(x|y)} = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)} \end{aligned} \tag{2.12}$$

$$\text{IM cond } (X,Y|Z) = H(X|Z) - H(X|Y,Z) \tag{2.13}$$

Tang et al [118] évaluent également l'indépendance mutuelle entre les deux vues grâce aux statistiques conditionnelles de l'information mutuelle conditionnelle et du test du χ^2, et proposent les algorithmes PMID-MI et PMID-chi comme méthodes de sélection de variables.

Ling et al [81] proposent une méthode de séparation des vues automatique basée sur l'entropie détaillée dans l'algorithme 4. Cet algorithme se base sur le principe que si une

5. `http://www.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb/`

variable a une entropie élevée, elle est très informative, et le fait de répartir les variables en les ayant trié par ordre d'entropie fait que les deux vues seront capables de fournir une prédiction.

Algorithme 4 : Algorithme de séparation en deux vues [81]

Entrée : $\mathcal{D}$
Sortie : $\mathcal{X}_1$, $\mathcal{X}_2$

```
calcul de l'entropie de chaque variable par rapport à D
tri des variables par entropie décroissante
pour chaque variable d'indice pair faire
    ajouter à X1
pour chaque variable d'indice impair faire
    ajouter à X2
```

2.4 Conclusion

Le co-training a été introduit comme une technique d'apprentissage conjointe de deux classifieurs dans un contexte semi-supervisé, et c'est toujours très majoritairement de cette façon qu'il est employé. Le co-training est supposé améliorer les performances par rapport à deux apprentissages simples si les hypothèses d'indépendance et de suffisance sont respectées.

De très nombreux travaux traitant de co-training portent sur l'étude de l'impact de chacune de ces deux conditions. Il n'y a pas consensus sur l'importance de l'hypothèse d'indépendance ; de nombreuses définitions existent avec de plus ou moins grandes exigences sur l'intensité de cette indépendance. Une étude propose notamment deux indicateurs δ_1 et δ_2 pour quantifier le respect de chacune des deux hypothèses. Les mesures de δ_1 et δ_2 sont de bons indicateurs à utiliser pour décider si le co-training est une option intéressante pour apprendre un modèle sur un jeu de données en particulier, ou encore pour comparer les performances d'une stratégie de co-training appliquée sur plusieurs jeux de données.

En résumé

- ✓ Le principe du co-training est d'entraîner deux modèles ensemble avec deux vues d'un jeu de données ;
- ✓ Il existe des méthodes pour créer deux vues avec un jeu de données lorsque celles-ci n'existent pas naturellement ;
- ✓ Le co-training repose sur des hypothèses de suffisance et d'indépendance mais il n'y a pas consensus sur l'importance de cette dernière ;
- ✓ Le co-training est majoritairement utilisé dans un contexte semi-supervisé ;

Deuxième partie

Contributions

Chapitre 3

Réseaux bayésiens pour le diagnostic à partir de capteurs

Dans un contexte de diagnostic industriel, des capteurs sont placés sur les parties de la machine à diagnostiquer dans le but de détecter des événements pouvant modifier leur état. Ce chapitre présente une architecture de réseau bayésien générique, répondant à une problématique de diagnostic détaillée section 3.1. La structure du réseau bayésien et ses différentes spécificités d'adaptation au problème du diagnostic industriel sont présentées dans la section 3.2. La section 3.3, quant à elle, détaille la méthode d'apprentissage des paramètres du modèle, ainsi que le mécanisme de mesure et de sélection de la pertinence

des entrées. La section 3.4 expose la création d'un ensemble de données artificielles adaptée pour la structure présentée précédemment et les expérimentations effectuées sur ces dernières pour valider la proposition précédente.

3.1 Formalisation du problème

La problématique sur laquelle se penche ce chapitre est détaillée en deux sous-sections, qui présentent d'abord le type de données de départ puis amènent vers les informations recherchées spécifiquement en terme de diagnostic.

3.1.1 Pré-traitement

Le pré-traitement présenté dans cette section concerne la transformation d'une série de données temporelles collectées par un capteur en une suite de points grâce à une agrégation par période. La formule générale est détaillée dans la première sous-section, puis la deuxième développe un exemple d'utilisation et d'interprétation des résultats.

Cas général

La suite des éléments de ce chapitre se base sur le fait que si l'état se dégrade, la valeur des capteurs a tendance à augmenter. En effet, pour un bon nombre de ces capteurs, des valeurs considérées comme normales et non révélatrices de problèmes correspondent à des valeurs inférieures à certains seuils. Symétriquement, des valeurs élevées ont tendance dans un grand nombre de cas à signifier un problème.

De plus, si ces capteurs sont mesurés en continu, il est généralement nécessaire d'utiliser une ou même plusieurs agrégations pour pouvoir exploiter la richesse de ces données, et pouvoir détecter notamment différents types d'incidents qui peuvent mener à des dégradations. En effet, certaines détériorations vont se déclencher après une vibration longue et pas nécessairement intense : le niveau des capteurs reste en-dessous d'un certain seuil τ_0. D'autres dégâts ne se feront que si un choc violent et très ponctuel se produit, c'est-à-dire quand le niveau des capteurs dépasse τ_1.

Godreau et al [41] proposent un critère de surveillance avancée X^τ, qui permet d'agréger les données quotidiennes $X(t)$ de capteurs sur une électrobroche selon un seuil τ. La formule pour le calcul de ce critère se trouve dans l'équation 3.1, où t_{max} est le temps sur lequel les valeurs vont être agrégés. Dans l'exemple illustré dans ces travaux, les valeurs

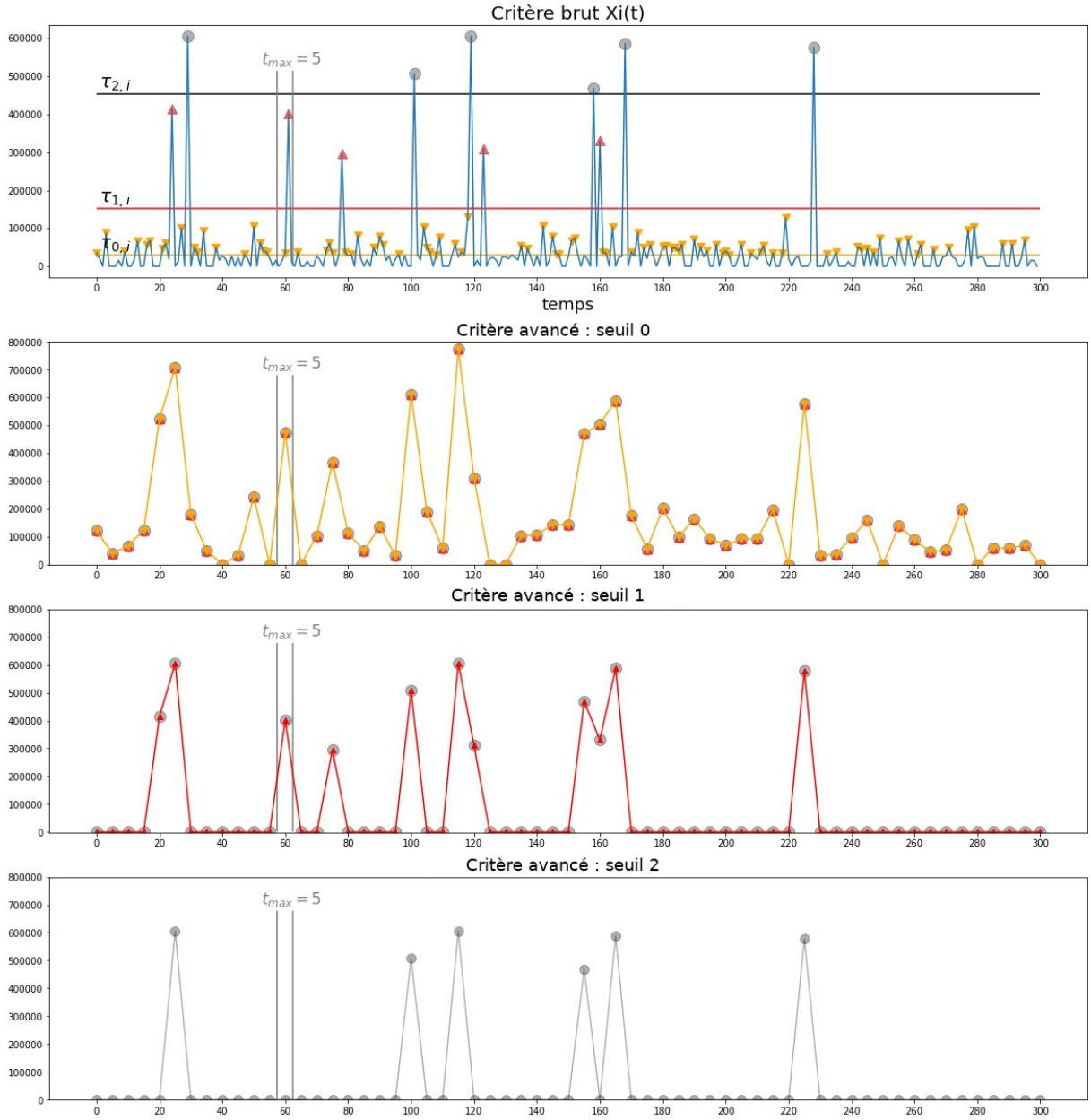


FIGURE 3.1 – Illustration du seuillage d'un critère brut. De haut en bas : graphe du critère brut, et résultat du calcul des critères avancés avec les seuils $\tau_{0,i}$, $\tau_{1,i}$ et $\tau_{2,i}$ à partir de la formule de l'équation 3.1. Les seuils sont indiqués respectivement par la couleur orange, rouge et noire, et les points de la même couleur sont les points dont la valeur est supérieure au seuil associé et inférieur au seuil supérieur. Les deux traits gris verticaux illustrent la période de $t_{max} = 5$.

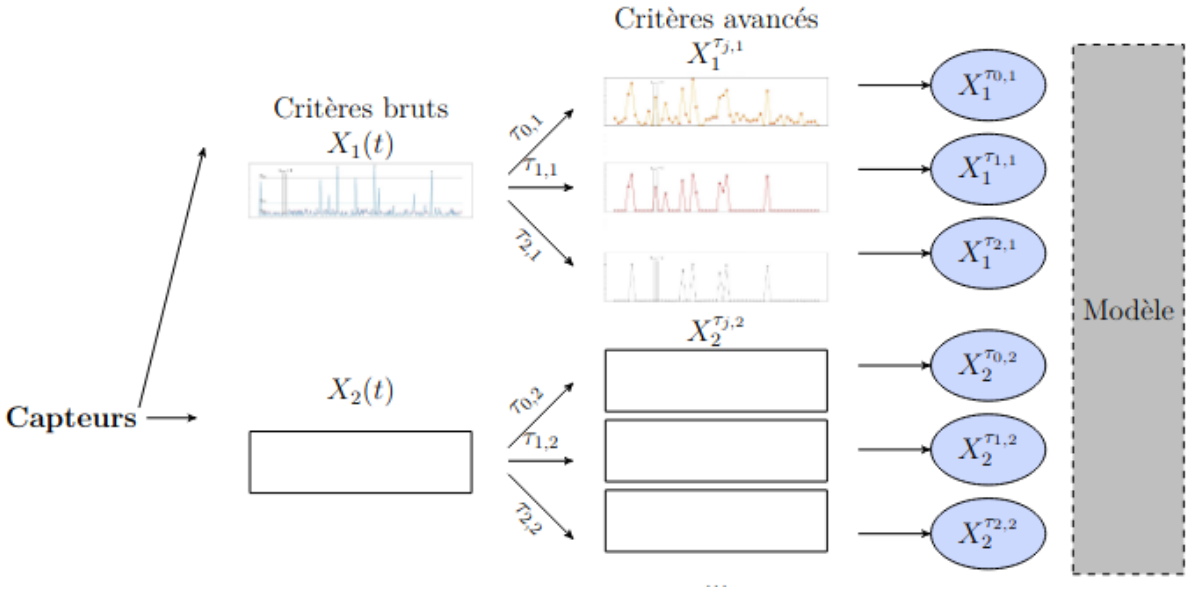


FIGURE 3.2 – Schéma de la chaîne globale de traitement des données, du capteur jusqu'à l'entrée dans le modèle.

sont agrégées sur une journée à partir de signaux échantillonnés tous les 10ème de seconde, donc $t_{max} = 86400$.

$$X^{\tau} = \sum_{t=1}^{t=t_{max}} \max(X(t) - \tau, 0) \tag{3.1}$$

L'avantage de cette formule est qu'elle permet de faire ressortir différents types d'incidents, qu'ils se produisent sur des durées courtes ou longues. Une valeur de τ élevée met en valeur les événements courts et violents, tandis qu'une valeur plus faible fait ressortir les événements longs et d'intensité plus modérée.

Exemple

Les données temporelles associées à cet exemple se trouvent illustrées dans la figure 3.1. Il y apparaît que ces données comprennent certains événements longs mais peu intenses, par exemple entre les temps $t = 180$ et $t = 200$, et ceux-ci ressortent sur la figure issue du seuil 0 (courbe orange) de la figure 3.1, mais sont absents des figures des seuils 1 et 2.

En revanche, les événements qui ressortent de la courbe noire du seuil 2 sont uniquement les événements ponctuels avec les valeurs les plus élevées, tels que ceux apparaissant aux temps $t = 29$, $t = 119$ et $t = 228$ sur la figure 3.1.

Cette suite de points correspond aux données utilisées en entrée par le modèle présenté dans les sections suivantes. Il peut être intéressant d'agréger les données avec plusieurs seuils différents pour faire ressortir différentes informations sur les types d'événements.

La figure 3.2 résume la façon dont ces différents critères avancés sont générés à partir des données de capteurs, puis passés en entrée du modèle.

3.1.2 Objectif du modèle

L'objectif premier du modèle est le diagnostic, celui-ci doit donc permettre de connaître l'état de la partie à diagnostiquer à un instant donné, en fonction d'un certain nombre de critères. Il doit précisément donner des informations sur les dégradations qu'a subi la partie surveillée dans une période de temps, comme défini dans la sous-section précédente.

De plus, il est difficile dans certains domaines industriels de connaître quels types d'incidents engendrent quels types de dégradation, ce qui est un frein à l'amélioration des procédés de fabrication. En effet, savoir que telle valeur de critère élevée entraîne un risque accru d'un endommagement spécifique peut permettre l'amélioration de la précision de la maintenance conditionnelle, et donc la limitations des coûts en intervention et matériel.

Dans ce cadre, il est souhaitable de construire un modèle qui prenne en compte ces différents types d'incidents pour améliorer la prédiction. Il serait également d'un grand intérêt d'avoir des indications sur l'importance d'un capteur donné et des seuils τ qui lui sont associés au moment de la prédiction, pour mieux comprendre l'impact des différents événements.

3.2 Modèle proposé

La figure 3.3 montre la structure générique proposée pour répondre au problème présenté dans la section 3.1. La discrétisation et les formes particulières des tables de probabilités sont décrites dans les sous-section suivantes.

3.2.1 Discrétisation

Comme discuté dans la section 1.3.2, considérer l'étape de discrétisation dans le modèle lui-même est une pratique courante lorsqu'il s'agit de modèles graphiques probabilistes : il s'agit alors d'un modèle hybride. C'est de cette façon que la discrétisation est traitée dans

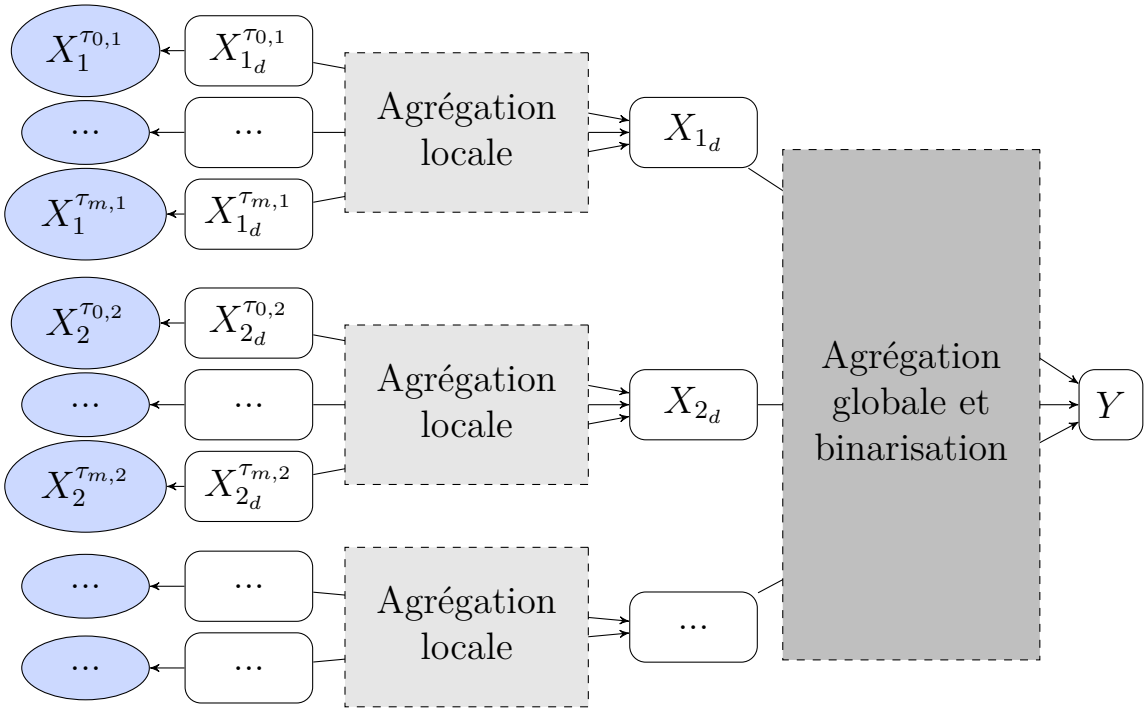


FIGURE 3.3 – Architecture générique de réseau bayésien. Les cercles représentent des variables aléatoires continues et les rectangles (arrondis) des variables discrètes. Les X_i sont les mesures de capteurs en entrée, $\tau_{m,i}$ étant le seuil mentionné dans la partie 3.1 et illustré dans la figure 3.1.

cette proposition, sous la forme d'un modèle de mélange gaussien dont les paramètres sont appris de manière non supervisée par l'algorithme EM (section 1.2.2).

Le modèle prend en entrée les variables continues $X_i^{\tau_{j,i}}$, à gauche de la figure 3.3. Ces variables représentent les critères de surveillance avancé décrit dans la section 3.1.1, et les données d'entrées de ces variables issues du pré-traitement de cette même sous-section. Chaque variable continue $X_i^{\tau_{j,i}}$ a un parent $X_{i_d}^{\tau_{j,i}}$ discrétisé en $r = 3$ intervalles. Ces intervalles de discrétisation du critère de surveillance avancé sont interprétables comme des niveaux de risque de dégradation croissant avec le rang de l'intervalle. Le nombre de 3 intervalles a été choisi car il a été déterminé dans une étude préliminaire [58] sur cette problématique qu'une discrétisation à seulement $r = 2$ intervalles faisait perdre une partie de l'information, qui correspond à la classe intermédiaires de problèmes peu grave.

3.2.2 Agrégation locale

Comme mentionné dans la section 3.1.1, la construction du modèle se base sur l'hypothèse que si l'état se dégrade, la valeur des critères a tendance à augmenter. Par conséquent, le rang de l'intervalle de discrétisation de chaque critère avancé donne une indication de l'état. A l'issue du pré-traitement des critères de surveillance brut X_i, un critère avancé $X_i^{\tau_{j,i}}$ a été obtenu pour chaque $\tau_{j,i}$. Comme décrit dans la section 3.1.1, chaque seuil critique $\tau_{j,i}$ donne des informations sur différents types d'événements.

Le but est d'agréger les discrétisations $X_{i_d}^{\tau_{1,i}}$ des critères avancés $X_i^{\tau_{j,i}}$ issus d'un même critère brut X_i pour obtenir une estimation locale X_{i_d} de l'état du système. Cette agrégation est nommée agrégation locale, indiquée sur la figure 3.3 dans un cadre gris clair, et deux méthodes sont proposées pour l'effectuer.

La première est l'opérateur très naïf max qui décrit le fait que la décision locale pour chaque critère est le niveau de problème le plus grave détecté parmi les critères avancés. Il n'est nécessaire de définir aucun paramètre pour cette agrégation ; $X_{i_d} = max(X_{i_d}^{\tau_{1,i}}, ..., X_{i_d}^{\tau_{m,i}})$.

La seconde est plus évoluée et consiste à utiliser une interaction canonique telle que la porte max bruité, décrite section 1.2.3. Avec ce type d'agrégation, la dépendance entre la décision et les entrées n'est plus déterministe, et l'influence de chaque entrée sur la sortie n'est plus constante. La forme proposée par Diez et Druzdzel [31] est utilisée, c'est-à-dire que $HX_{i_d}^{\tau_{1,i}}, ..., HX_{i_d}^{\tau_{m,i}}$ sont les paramètres bruités, ainsi $X_{i_d} = max(HX_{i_d}^{\tau_{1,i}}, ..., HX_{i_d}^{\tau_{m,i}})$. Les paramètres sont donc les $P(HX_{i_d}^{\tau_{j,i}} | X_{i_d}^{\tau_{j,i}})$ pour chaque i et j.

Chaque agrégation locale a donc transformé un ensemble d'évaluations basées sur la discrétisation $(X_{i_d}^{\tau_{1,i}}, ..., X_{i_d}^{\tau_{m,i}})$ en estimation locale de l'état du système pour chaque critère X_i, toujours basé sur l'hypothèse qu'une valeur élevée a tendance à signifier qu'il y a une dégradation de l'état.

3.2.3 Agrégation globale et binarisation

Après avoir obtenu un ensemble d'estimations locales X_{i_d} de l'état, il faut assembler ces estimations à l'aide d'une agrégation globale, indiquée sur la figure 3.3 dans un cadre gris foncé, pour obtenir un diagnostic global Y_d. Comme pour la décision locale de la section précédente, cette agrégation est traitée avec un max déterministe ou une porte max bruité, de la même façon que précédemment. L'issue de cette agrégation est une variable intermédiaire Y_d dont le domaine est le même que celui des critères précédents :

$Y_d = max(X_{1_d}, ..., X_{n_d})$ si l'agrégation globale est un max et $Y_d = max(HX_{1_d}, ..., HX_{n_d})$ si c'est un max bruité. Les paramètres sont donc $P(HX_{i_d}|X_{i_d})$ pour chaque i dans le cas d'un max bruité.

Enfin, l'issue de la décision globale est la valeur de la variable binaire Y, qui prend la valeur 0 s'il est estimé que la situation à un moment donné est en condition normale de fonctionnement (CNF), ou la valeur 1 s'il est en situation critique (KO). Cette décision est binarisée grâce à une distribution intermédiaire $P(Y|Y_d)$ décrite dans l'équation 3.2. Cette distribution est gelée et n'est pas modifiée au cours de l'apprentissage des paramètres. Elle fait ressortir le fait que seul un événement détecté comme grave à ce stade sera diagnostiqué comme ayant déclenché une dégradation (état KO).

$$P(Y|Y_d) = \begin{cases} 1 & \text{si } Y_d = r \\ 0 & \text{sinon.} \end{cases} \tag{3.2}$$

3.3 Apprentissage du modèle et utilisation

La section précédente détaillait les différents aspects du modèle proposé, la structure du graphe ainsi que le type de distribution de probabilités. La méthode d'apprentissage des paramètres du modèle est présentée dans cette section, ainsi que l'algorithme utilisé pour la sélection de variable, avec le détail des critères de sélection employés.

3.3.1 Apprentissage simple

L'apprentissage des paramètres de ce modèle peut se faire de manière supervisée ou non-supervisée. Le fait qu'il contienne des variables latentes rend obligatoire l'utilisation de l'algorithme EM. Comme mentionné section 1.2.2, l'initialisation est très importante pour l'apprentissage avec l'algorithme EM, celles-ci sont détaillées dans la première sous-section. De plus, pour une estimation des paramètres plus robuste qu'avec la méthode MLE (*maximum likelihood estimation*), la phase d'estimation des paramètres de EM (maximisation) se fait avec la méthode EAP (*expected a posteriori*). Dans ce contexte, des *a priori* de Dirichlet sont proposés pour chaque variable ou configuration de variables, comme détaillé par la suite.

	X_{i_d}		
$X_{i_d}^{\tau_{j,i}}$ ou $HX_{i_d}^{\tau_{j,i}}$	OK	dégradé	KO
OK	1	0	0
dégradé	0	1	0
KO	0	0	1

TABLE 3.1 – Table d'initialisation des valeurs pour $P(X_{i_d}|X_{i_d}^{\tau_{1,i}}, ..., X_{i_d}^{\tau_{m,i}})$ dans le cas du max ou $P(X_{i_d}|HX_{i_d}^{\tau_{1,i}}, ..., HX_{i_d}^{\tau_{m,i}})$ dans le cas de la porte max bruité. La table entière est grisée car cette distribution est gelée et n'est pas modifiée pendant l'apprentissage.

	HX		
X	OK	dégradé	KO
OK	1	0	0
dégradé	0	1	0
KO	0	0	1

TABLE 3.2 – Table d'initialisation des valeurs pour $P(HX|X)$ dans le cas d'une utilisation de porte max bruité. Plus spécifiquement, cette table représente la distribution de $P(HX_{i_d}^{\tau_{j,i}}|X_{i_d}^{\tau_{j,i}})$ dans le cas de l'agrégation locale ou $P(HX_{i_d}|X_{i_d})$ dans le cas de l'agrégation globale. La première ligne est grisée car cette partie de la distribution est gelée et n'est pas modifiée pendant l'apprentissage.

	Y_d		
X_{i_d} ou HX_{i_d}	OK	dégradé	KO
OK	1	0	0
dégradé	0	1	0
KO	0	0	1

TABLE 3.3 – Table d'initialisation des valeurs pour $P(Y_d|X)$ dans le cas du max ou $P(Y_d|HX)$ dans le cas de la porte max bruité. La table entière est grisée car cette distribution est gelée et n'est pas modifiée pendant l'apprentissage.

Initialisation des paramètres

Comme mentionné dans la section précédente, $X_{i_d} = max(X_{i_d}^{\tau_{1,i}}, ..., X_{i_d}^{\tau_{m,i}})$ si l'agrégation locale est un max, et $X_{i_d} = max(HX_{i_d}^{\tau_{1,i}}, ..., HX_{i_d}^{\tau_{m,i}})$ si c'est un max bruité. Les tables de probabilités $P(X_{i_d}|X_{i_d}^{\tau_{1,i}}, ..., X_{i_d}^{\tau_{m,i}})$ et $P(X_{i_d}|HX_{i_d}^{\tau_{1,i}}, ..., HX_{i_d}^{\tau_{m,i}})$ de ces deux cas de figures sont figées et présentées dans la table 3.1. Les paramètres sont les $P(HX_{i_d}^{\tau_{j,i}}|X_{i_d}^{\tau_{j,i}})$ pour chaque i et j et leur initialisation est celle de la table 3.2. La première ligne, $P(HX_{i_d}^{\tau_{j,i}}|X_{i_d}^{\tau_{j,i}} = OK)$, est grisée car cette partie de la distribution est gelée et n'est pas modifiée pendant l'apprentissage.

Le même principe s'applique avec l'agrégation globale, la table de probabilités de

	Y	
Y_d	OK	KO
OK	1	0
dégradé	1	0
KO	0	1

TABLE 3.4 – Table d'initialisation des valeurs pour $P(Y|Y_d)$. La table entière est grisée car cette distribution est gelée et n'est pas modifiée pendant l'apprentissage.

$P(Y_d|X_{1_d}, ..., X_{n_d})$ dans le cas du max, et celle de $P(Y_d|HX_{1_d}, ..., HX_{n_d})$ dans le cas d'une agrégation en max bruité, est figée et présentée dans la table 3.3.

Les paramètres sont donc $P(HX_{i_d}|X_{i_d})$ pour chaque i dans le cas d'un max bruité et leur initialisation est présentée dans la table 3.2. La première ligne, $P(HX_{i_d}|X_{i_d} = OK)$, est grisée car cette partie de la distribution est gelée et n'est pas modifiée pendant l'apprentissage.

Enfin, $P(Y|Y_d)$ est gelée et son initialisation, table 3.4, est sa forme finale.

A priori de Dirichlet

L'apprentissage des paramètres se fait avec l'algorithme EM, et une des propositions de ce chapitre est d'effectuer la phase d'estimation des paramètres (maximisation) avec la méthode EAP (*expected a posteriori*). En effet, celle-ci présente l'avantage d'être plus robuste que la méthode MLE (*maximum likelihood estimation*) lorsque la quantité de données est faible, et que celles-ci sont déséquilibrées. L'estimation des paramètres avec EAP (voir formule dans l'équation 1.6 de la section 1.2.2) nécessite la définition des *a priori* de Dirichlet, notés α. Ils correspondent à un nombre d'observations virtuelles de chaque valeur prise par une variable ou une configuration de variables. Le choix des α est délicat, car un nombre trop élevé par rapport au nombre total d'instances donnera une importance négligeable aux valeurs des données d'entrées au moment de l'apprentissage, tandis qu'un nombre trop faible risque d'être équivalent à ne pas avoir donné d'information au modèle.

Comme vu précédemment, la seule partie apprise des paramètres sont les probabilités de $P(HX_{i_d}^{\tau_{j,i}}|X_{i_d}^{\tau_{j,i}} =$ dégradé ou KO) dans le cas de l'agrégation locale en max bruité et $P(HX_{i_d}|X_{i_d} =$ dégradé ou KO) dans le cas de l'agrégation globale en max bruité. Ce sont donc les distributions concernées par un ajout d'information a priori. Plusieurs a priori, notés $\alpha^{deg} = [\alpha_0^{deg}\ \alpha_1^{deg}\ \alpha_2^{deg}]$ et $\alpha^{KO} = [\alpha_0^{KO}\ \alpha_1^{KO}\ \alpha_2^{KO}]$, sont envisagés pour assurer l'apprentissage de probabilités pertinentes. Le premier cas de figure est un a priori

uniforme à 1, qui présente l'avantage d'être simple. Cependant, les événements recherchés sont rares, et dans un contexte de classes très déséquilibrées, il est possible que le modèle ne détecte aucun KO si $\alpha_2^{KO} = 1$. Pour éviter cette situation, $\alpha_2^{KO} = 10$ et $\alpha_2^{KO} = 20$ sont donc également proposés. Ces valeurs peuvent donner au modèle une impulsion pour lui permettre de mieux identifier les quelques endommagements existants. Par ailleurs, il est intéressant de considérer des $\alpha_1^{deg} = 5$, en supplément des $\alpha_1^{deg} = 1$, pour que le modèle soit encouragé à retrouver des événements dans les trois catégories OK, dégradé, et KO, et pas seulement dans OK et KO.

Au total, cinq options, qui résultent des combinaisons de $\alpha_2^{deg} = 1$ ou 5, et $\alpha_2^{KO} = 10$ ou 20 mentionnés précédemment sont étudiées :

— $\alpha^{deg} = [1\ 1\ 1]$ et $\alpha^{KO} = [1\ 1\ 1]$
— $\alpha^{deg} = [1\ 5\ 1]$ et $\alpha^{KO} = [1\ 1\ 10]$
— $\alpha^{deg} = [1\ 5\ 5]$ et $\alpha^{KO} = [1\ 1\ 10]$
— $\alpha^{deg} = [1\ 5\ 1]$ et $\alpha^{KO} = [1\ 1\ 20]$
— $\alpha^{deg} = [1\ 5\ 5]$ et $\alpha^{KO} = [1\ 1\ 20]$

3.3.2 Mesure de pertinence des entrées

Un des objectifs du modèle est d'aider à la compréhension des causes des dégradations. L'information mutuelle est une mesure de la quantité d'information commune entre deux variables, et peut donner une indication de la contribution d'une variable d'entrée sur la sortie d'un classifieur [75]. La méthode de calcul utilisée est indiquée dans l'équation 3.3, où H est l'entropie.

$$IM(X, Y) = H(X) + H(Y) - H(X; Y) \tag{3.3}$$

Le calcul de l'information mutuelle $IM(X_i^{\tau_{j,i}}, Y)$ entre chaque critère de surveillance avancé et l'état Y donne ainsi une indication sur la pertinence de chaque critère avancé.

Cependant, les variables issues des différentes valeurs de seuils pour un critère peuvent être corrélées. Le calcul de l'information mutuel est très contextuel, et lorsque les entrées sont corrélées, le modèle peut accorder de l'importance à des variables moyennement pertinentes. Dans ce contexte, tirer des conclusions à partir des informations mutuelles les plus élevées parmi toutes les entrées risque d'induire en erreur. Pour éviter cela, il est préférable d'adopter une approche de sélection de variables par élimination, qui permet à l'inverse de supprimer les variables les moins pertinentes.

Algorithme 5 : Sélection de variables par suppression de variables

Entrée : Ensemble de variables $\mathbf{X}$
Sortie : Ensemble de variables pertinentes $\mathbf{X}^p$

$\mathbf{X}_s = \mathbf{X}$
tant que *le coude dans l'évolution de l'IM n'est pas détecté* **faire**
    `// Apprentissage des paramètres du modèle avec toutes les variables de` $\mathbf{X}_s$
    $\theta = argmax\ P(\mathbf{X}_s)$
    pour *tous les seuils de tous les critères dans* $\mathbf{X}_s$ **faire**
        Calcul de $IM(X_i^{\tau_{j,i}}, Y)$
    $X_{min} = argmin(IM(X_i^{\tau_{j,i}}, Y))$
    $\mathbf{X}_s = \mathbf{X}_s \backslash X_{min}$

3.3.3 Sélection des entrées pertinentes

Pour reprendre les catégories de la section 1.4.2, la méthode de sélection est ici une approche par filtre. La sélection des entrées est faite par élimination (*backward elimination*) et consiste à apprendre le modèle en premier lieu avec toutes les variables, puis de les supprimer une à une à partir d'un critère donné. Le fait de supprimer en premier lieu les variables avec l'information mutuelle la plus faible permet d'éviter le problème cité précédemment, c'est-à-dire la sélection erronée en choisissant les variables avec l'information mutuelle la plus élevée (sélection de variables par ajout, *forward selection*).

L'algorithme 5 présente le détail de l'algorithme : le modèle est appris avec tous les critères de surveillance avancé, et l'information mutuelle $IM(X_i^{\tau_{j,i}}, Y)$ entre chaque critère de surveillance avancé et l'état Y est calculée.

La valeur de seuil ayant l'information mutuelle la plus faible est déterminée comme étant la moins informative et elle est supprimée du modèle. La sélection se poursuit jusqu'à la détection d'un coude dans l'évolution de l'information mutuelle, à l'aide de l'algorithme Kneedle de Satopää et al [112], détaillé dans la section 1.4.2.

3.4 Expériences

L'objectif des expériences est de déterminer s'il est possible grâce à la structure présentée de retrouver l'influence de chaque critère et chaque seuil sur la prédiction finale.

Plusieurs jeux de données ont été créés dans un environnement très contrôlé, présenté succinctement section 3.4.1 et détaillé dans l'annexe 5.6, pour pouvoir comparer la façon

jdd	21	22	23	24	25	26	27	28	29
r_D	0.08	0.2	0.12	0.19	0.04	0.01	0.02	0.02	0.06

TABLE 3.5 – Taux de déséquilibre r_D pour chaque jeu de données. Il s'agit du nombre de KO divisé par 300 (nombre de lignes total).

dont sont déclenchés artificiellement les incidents dans les données, et ce qui est retrouvé par le modèle. La stratégie d'apprentissage utilisée est exposée dans la partie 3.4.2, et les résultats des expériences sont présentés dans la partie 3.4.3.

3.4.1 Simulation de données

Aucun jeu de données public n'a pu être identifié pour la validation du modèle proposé. Cela peut être dû au fait que les données de production sont souvent conservées privées. Pour la partie expérimentale de cette contribution, il a donc été nécessaire de créer des jeux de données artificiels. La procédure détaillée se trouve dans la section 5.6 en annexe.

Neuf jeux de données artificiels sont évalués ici, quatre d'entre eux (**22**, **23**, **24** et **25**) sont influencés par un seuil d'un seul critère, respectivement le 3ème seuil du 2ème critère, le 3ème seuil du 1er critère, le 1er seuil du 4ème critère, le 2ème seuil du 4ème critère.

Deux autres sont influencés par 2 seuils du même critère simultanément : les 2ème et 3ème seuils du 4ème critère pour le jeu de données **26**, les 1er et le 3ème seuils du 4ème critère pour le **29**. Deux jeux de données sont influencés par deux seuils de deux critères différents : le **27** avec le 1er seuil du 1er et du 4ème critère, et le **28** le 1er et le 3ème seuil du 3ème critère. Enfin, un dernier jeu de données plus complexes, le **21**, influencé par quatre seuils est évalué.

Chaque jeu de données comporte 300 lignes, et la table 3.5 donne les taux de déséquilibre des jeux de données générés, c'est-à-dire la part de KO pour le nombre de lignes total (nombre de KO/300).

3.4.2 Protocole expérimental

L'apprentissage des paramètres du modèle se fait de façon supervisée. L'agrégation locale et l'agrégation globales utilisées sont respectivement le max bruité et le max. Ce deuxième choix est dû à une limite de notre implémentation actuelle qui rend impossible l'apprentissage du modèle avec toutes les agrégations en max bruité.

Enfin, les expériences sont réalisées avec cinq validations croisées à 2 échantillons (*2-fold cross-validation*).

3.4.3 Résultats

Les résultats présentés détaillent l'ordre de suppression des variables obtenu avec l'algorithme 5, et l'impact qu'a cette sélection sur les performances du classifieur.

Détermination des a priori de Dirichlet

Dans le cas d'un jeu de données déséquilibré, utiliser des a priori de Dirichlet peut permettre de mieux apprendre les paramètres du modèle. Des expériences ont été effectuées avec la stratégie d'apprentissage supervisée pour déterminer quel a priori est le plus adapté sur les données générées parmi les cinq combinaisons de α^{deg} et α^{KO} listées dans la section 3.3.1.

La table 3.6 montre les différences de performances moyennes et leur écart type, les performances minimum et maximum atteintes pendant les différentes itérations de la sélection de variable. La table 3.7 permet de savoir lesquelles de ces valeurs sont significativement meilleures que les autres en précision équilibrée et sensibilité, à l'aide d'un test de Student. Dans cette table, toutes les cases de la colonne u sont vertes, ce qui signifie que l'*a priori* uniforme avec tous les α à 1 est significativement meilleur que les autres *a priori*. Par conséquent, pour la suite de ce chapitre, $\alpha^{deg} = [1\ 1\ 1]$ et $\alpha^{KO} = [1\ 1\ 1]$.

	Précision équilibrée (%)				**Sensibilité (%)**			
	moyenne	**éc. type**	**min**	**max**	**moyenne**	**éc. type**	**min**	**max**
u	89.5	12.6	46.6	100	85	25	0	100
1,10	81.1	18.2	40.2	99.3	75.5	35.6	0	100
1,20	78.1	19.1	39.9	98.8	70.1	37.7	0	100
5,10	84.2	13.3	39.6	99.7	85.6	27	0	100
5,20	82.3	15.1	40.2	98.6	81.9	31.2	0	100

TABLE 3.6 – Performances atteintes respectivement en précision équilibrée et sensibilité avec la stratégie supervisée selon les différentes paires de valeurs de α_2 pour dégradé et KO. La lettre u désigne les résultats pour les entraînements avec un a priori uniforme de 1 pour tous les α. Pour les autres a priori, le premier nombre est α_2^{deg} et le deuxième α_2^{KO}.

Identification des variables importantes

La table 3.8 présente l'ordre inverse moyen de suppression des variables avec l'algorithme 5 présenté dans la sous-section 3.3.3. Les cases colorées représentent les variables influentes au moment de la création du jeu de données. Une case est colorée en vert si la valeur est inférieure ou égale à cinq, et en vert foncé si la valeur est inférieure ou égal à trois.

Précision équilibrée

	u	**1,10**	**1,20**	**5,10**	**5,20**
u					
1,10					
1,20					
5,10					
5,20					

Sensibilité

	u	**1,10**	**1,20**	**5,10**	**5,20**
u					
1,10					
1,20					
5,10					
5,20					

TABLE 3.7 – Significativité des comparaisons entre les différentes paires de valeurs de α_2 pour dégradé et KO, obtenu avec un test de Student à un risque de 1%. La lettre u désigne les résultats pour les entraînements avec un a priori uniforme de 1 pour tous les α. Pour les autres a priori, le premier nombre est α_2 pour dégradé et le deuxième α_2 pour KO. La table de gauche indique les différences significatives pour la précision équilibrée, et la table de gauche pour la sensibilité. Vert et rouge signifient respectivement que la valeur en colonne est significativement meilleure ou pire que la valeur en ligne. Les cases grises sont les cases dont les valeurs sont non significatives.

	s11	**s21**	**s31**	**s12**	**s22**	**s32**	**s13**	**s23**	**s33**	**s14**	**s24**	**s34**
21	5	6	2	4	10	1	8	11	0	7	9	3
22	7	10	12	6	3	1	2	11	4	9	8	5
23	7	11	4	1	2	6	2	5	12	8	8	10
24	4	6	8	5	10	12	7	9	11	1	2	3
25	5	7	8	3	4	8	10	12	11	1	2	6
26	10	11	12	8	7	9	4	3	6	1	2	5
27	1	3	5	9	10	12	4	8	11	2	6	7
28	3	5	7	4	10	12	1	6	2	8	9	11
29	5	7	10	3	6	8	9	11	12	2	4	1

TABLE 3.8 – Ordre inverse moyen de suppression des variables de l'ensemble des critères d'apprentissage : 1 signifie que la variable est supprimée en dernier en moyenne, c'est-à-dire qu'elle est considérée comme la plus importante. Les cases colorées représentent les variables influentes au moment de la création du jeu de données. Une case est colorée en vert si la valeur est inférieure ou égale à cinq, et en vert foncé si la valeur est inférieure ou égal à trois (*i.e.* dans le top 3 des valeurs les plus importantes pour la prédiction).

Les seuils d'influence présentés dans la section 3.4.1 pour chaque jeu de données sont donc ceux que l'on souhaite retrouver dans la table 3.8 avec les indices les plus faibles.

Dans cinq des jeux de données (**22**, **24**, **27**, **28**, **29**), au minimum une des variables influentes a obtenu le score 1, c'est-à-dire qu'elle a été retrouvée comme la plus importante au moment de la prédiction. Au total, sur les 16 variables à retrouver, 11 ont un score inférieur ou égal à trois, et 15 ont un score inférieur ou égal à 5. Seule une variable a obtenu un score supérieur à 5, cependant, il s'agit d'une des quatre variables influentes du jeu de donnés **21**.

Il est important de souligner que les erreurs répertoriées sont en général des erreurs sur le seuil, et non sur le critère. Par exemple, s34 (3ème seuil du critère 4) qui est une des deux variables influentes de **26**, n'obtient un score que de 5. Cependant, c'est la variable s14 (1er seuil pour le même critère) qui obtient la première place, et s24 (2ème seuil) la deuxième. En revanche, **29** qui est dans le même cas (deux seuils différents pour le même critère) retrouve bien s14 et s34 avec les meilleurs scores (1 et 2). Il ressort qu'il est plus difficile pour l'algorithme de retrouver les deux variables lorsque celles-ci sont issues du même critère.

Pour le jeu de données plus complexe, **21**, influencé par quatre variables, trois d'entre elles sont retrouvées avec des scores inférieurs à 5, mais l'une d'entre elles a un score de 8. Par ailleurs, il est notable que les variables obtenant les trois meilleurs score pour ce jeu de données (s33 qui n'est jamais supprimée et obtient donc un score de 0, s32 avec 1 et s31 avec 2) ne sont pas des variables influentes. Ce sont toutes les trois des variables qui sont issues d'un seuil différent des variables originales, ce qui fait écho à la remarque précédente : il apparaît que l'algorithme peut avoir des difficultés à distinguer certains seuils, notamment en faveur d'un seuil plus élevé.

Dans ces expériences, les cinq variables restantes à la fin de l'algorithme contiennent toujours les variables recherchées, pour les jeux de données ayant jusqu'à deux variables d'influence. Le fait de garder cinq variables pour avoir un modèle intéressant n'est pas propre à cette étude et Hoque et al [54] évoque également ce nombre comme un minimum de variables à conserver sous peine de pénaliser la précision du classifieur, bien que cette étude utilise un modèle différent.

Importance des variables dans les prédictions

La figure 3.4 montre l'évolution de la précision équilibrée et de la sensibilité selon les itérations de l'algorithme de sélection de variables pour les jeux de données **22**, **25** et **26**,

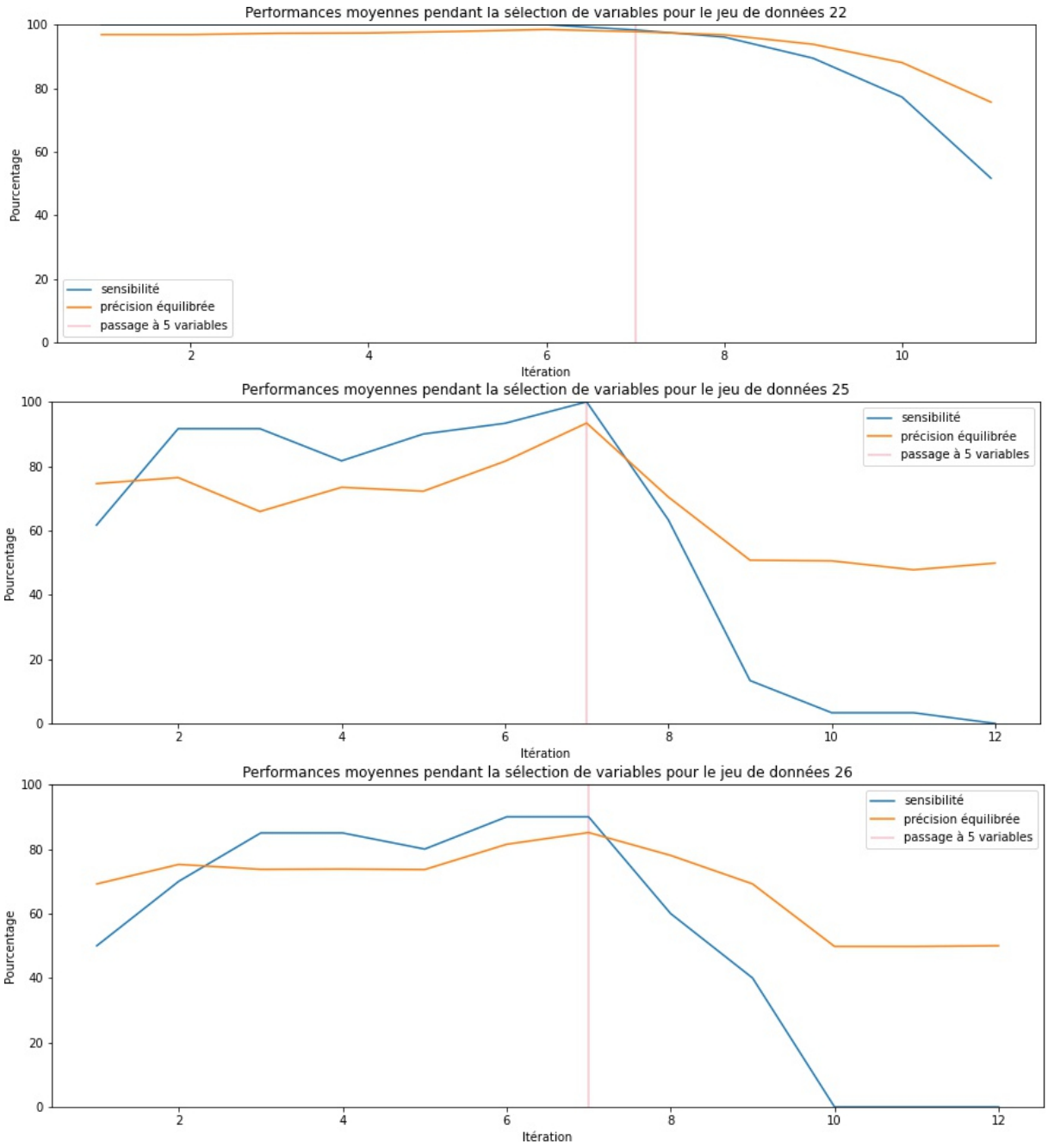


FIGURE 3.4 – Performances moyennes pendant la sélection de variables pour le jeu de données **21**,**25** et **26**. Le trait vertical rose indique le passage à cinq variables pour l'apprentissage, qui marque une chute de performances.

	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
précision équilibrée (en %)	84	93	85	95	92	81	96	96	95
sensibilité (en %)	100	100	90	100	100	90	100	100	100

TABLE 3.9 – Meilleures performances atteintes respectivement en précision équilibrée et sensibilité pour chaque jeu de données avec la stratégie supervisée.

et mettent en valeur cette chute de précision lorsque le sous-ensemble passe en dessous de cinq variables. De plus, ces figures montrent distinctement deux types d'évolution différentes pour les performances. Le jeu de données **21** est influencé par un seuil de chacun des quatre critères, et il est visible que toutes les variables aident à la prédiction dès le début, et qu'aucune de celles qui sont enlevées jusqu'à l'itération 7 n'apporte des informations indispensables (les performances sont stables). En revanche, dans le cas des jeux de données **22** et **23** qui sont influencés par une seule variable (bien qu'elles soient également corrélées avec les variables issues du même critère), les performances sont mauvaises au début car les critères inutiles apportent du bruit à la prédiction. Cependant, l'itération 7 (indiquée par le trait rose sur la figure, et qui correspond au passage à cinq variables) indique le maximum de performances, dû au fait que toutes les variables inutiles ont été supprimées auparavant, et que toutes les variables supprimées ensuite aident à la prédiction.

Les performances du classifieur sur ces jeux de données varient selon les itérations en fonction du nombre de variables, et les performances maximum atteintes par le classifieur pour chaque jeu de données sont récapitulées table 3.9. Ces valeurs montrent les bonnes capacités de prédiction du classifieur sur des jeux de données d'une complexité relativement limitée.

3.5 Conclusion

Ce chapitre propose un réseau bayésien adapté au diagnostic à partir de capteurs et une méthode d'apprentissage de ce modèle, et avance également une procédure de sélection des critères importants pour un problème de diagnostic donné. Ce modèle est générique et utilisable dans un contexte supervisé ou non-supervisé.

Cet ensemble de contributions est testé de manière supervisée dans la section expérimentale, qui prouve la capacité de l'algorithme de sélection à identifier les variables les plus utiles pour la prédiction, ainsi que ses performances en tant que classifieur, et ce

même sur des jeux de données très déséquilibrés, tels que les jeux de données 26, 27 et 28 qui n'ont respectivement que 4, 5 et 6 KO.

Une des limites de ces résultats est la complexité limitée des jeux de données. De ce fait, il serait intéressant de tester la capacité du modèle à retrouver les variables pertinentes grâce à des jeux de données plus complexes, avec plus de critères et plus de seuils.

Le fait que le modèle soit appris avec une agrégation globale en max limite possiblement les performances du classifieur et est dû à des contraintes de complexité dans l'implémentation. Il serait possible d'apprendre dans un premier temps le modèle et d'utiliser l'algorithme de sélection de variables tel que présenté dans la section précédente, puis d'utiliser le classifieur avec les variables sélectionnées et une agrégation globale en max bruité.

Une autre amélioration de ce travail pourrait être d'ajouter la mesure d'information mutuelle entre les variables d'entrée, tel que proposé par Battiti [8] ou Hoque et al [54], pour éviter les erreurs citées dans la section 3.4.3 où certains seuils sont sélectionnés à la place d'autres.

De plus, le contexte des expériences ne prouve la capacité du modèle à sélectionner les variables que dans un contexte supervisé. Il serait intéressant de tester les mêmes conditions expérimentales avec un apprentissage non-supervisé, et de comparer l'ordre d'élimination des variables entre les deux types d'apprentissage.

En résumé

- ✓ L'algorithme de sélection de variables permet la sélection des variables les plus utiles à la prédiction.
- ✓ Les performances du classifieur sont élevées pour les jeux de données générés utilisés, malgré un déséquilibre des classes très important pour certains.
- ✓ Le réseau bayésien proposé et la méthode d'apprentissage sont prometteuses pour l'utiliser en application réelle pour le diagnostic à partir de capteurs dans un contexte industriel.

Chapitre 4

Co-training non-supervisé

Le chapitre 2 a montré que dans de très nombreux cas l'utilisation du co-training semi-supervisé permet d'améliorer les performances de chaque vue. Cependant, le co-training non-supervisé reste largement inexploré et pourrait avoir une utilité dans certaines applications où il est difficile d'obtenir des données annotées. Ce chapitre présente trois stratégies non-supervisées de co-training pour la classification détaillées dans la section 4.1. Les stratégies présentées sont testées expérimentalement sur cinq jeux de données UCI adaptés pour le co-training, et quatre jeux de données artificiels, en utilisant pour l'apprentissage le classifieur bayésien naïf. Les résultats et les conclusions de ces expériences sont développées section 4.2.

4.1 Stratégies proposées

Cette section propose trois stratégies génériques de co-training non-supervisé. Ces stratégies sont applicables sur tout classifieur probabiliste.

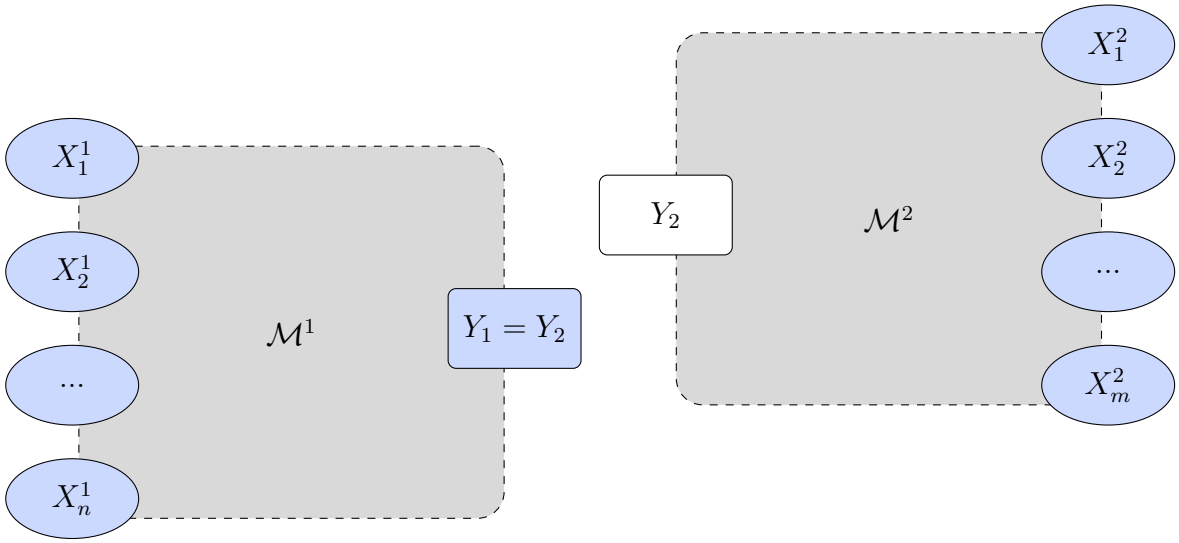


FIGURE 4.1 – Illustration du comportement des modèles pendant la stratégie de co-training *split*. Les nœuds bleus correspondent aux variables observées et les nœuds blancs aux variables non observées. $\mathcal{M}_2$ est tout d'abord appris de façon non-supervisée, puis $\mathcal{M}_1$ est appris de façon supervisée, en reprenant la prédiction Y_2 de $\mathcal{M}_2$ pour l'assigner à Y_1.

4.1.1 Stratégie *split*

La stratégie *split* propose d'apprendre tout d'abord entièrement un seul des deux modèles, de façon non-supervisée à l'aide de l'algorithme EM. Suite à cet apprentissage, les prédictions de la classe sont obtenus par inférence probabiliste et application de la règle de Bayes $y^* = argmax P(y|\mathbf{X})$. Ces prédictions sont ensuite intégrées comme étant la vérité terrain pour le deuxième modèle, et celui-ci est appris de façon supervisée avec cette vérité estimée. La procédure est détaillée dans l'algorithme 6 où $P(\mathcal{X}_1|\theta_1)$ est la probabilité d'observer les données $\mathcal{X}_1$ pour le modèle $\mathcal{M}_1$, et symétriquement pour le modèle $\mathcal{M}_2$.

La figure 4.1 illustre ce fonctionnement avec un exemple de deux modèles symétriques

Algorithme 6 : Stratégie *split*

Entrée : $\mathcal{X}^1$, $\mathcal{X}^2$
Sortie : θ_1^*, θ_2^*
1 $\theta_2^* = argmax\ P(\mathcal{X}^2|\theta_2)$
2 $\mathbf{y}_2^* = argmax\ P(\mathbf{y}_2|\mathcal{X}^2, \theta_2^*)$
3 $\theta_1^* = argmax\ P(\mathcal{X}^1, \mathbf{y}_1 = \mathbf{y}_2^*|\theta_1)$

appris par la stratégie *split*, en commençant par le modèle $\mathcal{M}_2$. Dans ce cas, $\mathcal{M}_1$ profite de l'apprentissage non-supervisé de $\mathcal{M}_2$. Cependant, de manière symétrique, un meilleur apprentissage de $\mathcal{M}_1$ aiderait également à l'apprentissage de $\mathcal{M}_2$, et la stratégie *split* n'impose pas de contrainte sur le modèle appris le premier.

La stratégie *split* est la plus simple ayant pu être imaginée dans le contexte d'un apprentissage non-supervisé, et la stratégie *recursive* est dans la continuité de cette approche.

4.1.2 Stratégie *recursive*

La stratégie *recursive* est dans la continuité de *split*, en ajoutant l'aspect itératif de l'apprentissage tel que proposé par Blum et Mitchell [14] ou Nigam et Ghani [96]. et consiste à poursuivre l'apprentissage pour plusieurs itérations. Le début de l'algorithme *recursive* est donc exactement *split*, mais ne s'arrête pas après le premier apprentissage supervisé du deuxième modèle.

Pour reprendre l'exemple de l'illustration 4.1, après l'apprentissage puis l'obtention du résultat de l'inférence probabiliste du modèle $\mathcal{M}_1$, $\mathcal{M}_2$ est appris de nouveau en considérant comme vérité la prédiction de $\mathcal{M}_1$, etc. Comme le montre l'algorithme 7, le nombre d'itération n_{step} total est prédéfini comme hyper-paramètre à l'algorithme.

Dans la stratégie *recursive*, le vecteur de prédiction utilisé est entièrement renouvelé à chaque itération. Ce choix a été fait pour ne pas trop contraindre les modèles à l'apprentissage, et les laisser progresser dans la même direction. En théorie, si l'apprentissage a fonctionné comme prévu, les prédictions des modèles devraient tendre naturellement vers un optimum commun.

La stratégie *progressive* construit le vecteur de vérité de manière plus graduelle.

Algorithme 7 : Stratégie *recursive*

Entrée : $\mathcal{X}^1$, $\mathcal{X}^2$, n_{step}
Sortie : θ_1^*, θ_2^*

$\theta_2^* = \ argmax\ P(\mathcal{X}^2|\theta_2)$
pour *i allant de 1 à* n_{step} **faire**
  $\mathbf{y}_2^* \ = \ argmax\ P(\mathbf{y}_2|\mathcal{X}^2, \theta_2^*)$
  $\theta_1^* \ = \ argmax\ P(\mathcal{X}^1, \mathbf{y}_1 = \mathbf{y}_2^*|\theta_1)$
  $\mathbf{y}_1^* \ = \ argmax\ P(\mathbf{y}_1|\mathcal{X}^1, \theta_1^*)$
  $\theta_2^* \ = \ argmax\ P(\mathcal{X}^2, \mathbf{y}_2 = \mathbf{y}_1^*|\theta_2)$

Algorithme 8 : Stratégie progressive

Entrée : $\mathcal{X}^1$, $\mathcal{X}^2$, μ
Sortie : θ_1^*, θ_2^*

$\theta_2^* = \ argmax\ P(\mathcal{X}^2|\theta_2)$
$\mu_0 = 0$
$\mathbf{y}_0 = \{\}$
tant que *le critère d'arrêt n'est pas atteint* **faire**
  $\mathbf{y}_2^* = \ argmax\ P(\mathbf{y}_2|\mathcal{X}^2, \theta_2^*)$
  `// tri des éléments de` $\mathbf{y}_2$ `par ordre de vraisemblance décroissante`
  $\mathbf{y}_2^{trié} = \{l_1, l_2, ..., l_N\}$
  `// calcul du nombre d'instances` N_μ `à ajouter à` $\mathbf{y}$
  $N_\mu = \mu \times N$
  `// ajout des instances sélectionnées`
  $\mathbf{y} = \{l_k\}_{k=1}^{N_\mu}$
  $\mathbf{y}_i = \mathbf{y}_{i-1} \cup \mathbf{y}$
  $\theta_1^* = argmaxP(\mathcal{X}^1, \mathbf{y}_1 = \mathbf{y}_i|\theta_1)$
  $\mu_i = \mu_{i-1} + \mu$
  `// changement de vue`
  $\mathbf{y}_1^* = argmaxP(\mathbf{y}_1|\mathcal{X}^1, \theta_1^*)$

4.1.3 Stratégie *progressive recursive label selection*

La troisième stratégie proposée, *progressive recursive label selection* est dérivée de la stratégie *recursive*, mais avec la particularité de ne pas considérer comme vecteur de vérité la totalité des instances étiquetées dès la première itération. Ce principe est le même que celui proposé par Blum et Mitchell [14], à la différence qu'aucune distinction n'est faite selon la classe de la prédiction. En effet, un certain pourcentage μ du nombre total N est ajouté au vecteur de vérité de la classe à chaque itération. Comme le montre l'algorithme 8, à chaque itération $N_\mu = \mu \times N$ étiquettes sont ajoutées au vecteur de vérité. $\mu_i = \mu \times i$ correspond au pourcentage total d'instances du vecteur de vérité qui comporte une étiquette à une itération i donnée.

Les points à ajouter sont choisis en fonction d'une certaine mesure de confiance du modèle en l'annotation, tel que proposé par Nigam et Ghani [96]. La mesure de confiance utilisée dans l'algorithme proposé est ici la vraisemblance. L'intérêt de ne pas ajouter des étiquettes peu certaines dès le début de l'apprentissage est d'éviter d'orienter trop tôt les modèles dans une mauvaise direction, notamment lorsque les vues ne sont pas suffisantes, ce qui correspond au bruit d'étiquetage décrit par Wang et Zhou [133].

Contrairement à la stratégie *recursive*, le vecteur de vérité est désormais immuable ; chaque annotation qui y a été ajoutée ne pourra plus changer jusqu'à la fin. Le but de cette variation est un apprentissage plus graduel et donc une limitation du risque que les modèles s'induisent l'un l'autre en erreur dès le début de l'apprentissage. Cette stratégie s'approche plus d'un apprentissage semi-supervisé, comme dans le cas de nombreux algorithmes de co-training citées dans le chapitre 2.

Deux critères d'arrêt simple sont proposés pour cet algorithme. La première option est de poursuivre les itérations jusqu'à ce que toutes les instances soient étiquetées (p_m), et la deuxième est d'arrêter l'apprentissage lorsqu'un coude est détecté dans la confiance du modèle en les étiquettes qu'il ajoute (p_c).

4.2 Expériences

Pour cette partie expérimentale, les trois stratégies présentées précédemment dans la partie 4.1 sont utilisées pour entraîner deux classifieurs bayésiens naïfs $\mathcal{M}_1$ et $\mathcal{M}_2$. Les données et le protocole utilisés pour ces apprentissages sont présentés respectivement dans les sous-sections 4.2.1 et 4.2.2. Les résultats et leurs interprétations sont exposés dans la sous-section 4.2.3.

4.2.1 Données

Les jeux de données utilisés pour l'apprentissage sont les quatre jeux de données générés artificiellement, présentés dans la section 3.4.1, et des données extraites du dépôt UCI puis adaptées au co-training. Les caractéristiques intéressantes du point de vue du co-training, ainsi que des informations générales telles que le ratio de déséquilibre r_D sont présentées pour chacun de ces ensembles de données.

Données générées pour le co-training

Les données générées artificiellement présentées dans le chapitre précédent contenaient quatre critères avec chacun trois seuils, soit douze variables d'entrée au total pour la vue $\mathcal{X}_1$. Dans un souci de simplicité, seul un seuil a été conservé par critère, ce qui donne quatre variables pour la vue, $\mathcal{X}_1$, et ainsi que pour la vue $\mathcal{X}_2$ qui en contenait déjà quatre.

La table 4.1 présente pour chacun des quatre ensembles de données le nombre de variables chaque vue $\mathbf{X}_1$ et $\mathbf{X}_2$, le nombre d'instances N et le taux de déséquilibre r_D,

	δ_1	δ_2	$\lvert\mathbf{X}_1\rvert$	$\lvert\mathbf{X}_2\rvert$	N	r_D
jdd1 (21)	0.13	0.20	4	4	300	8%
jdd2 (22)	0.17	0.24	4	4	300	20%
jdd3 (23)	0.10	0.39	4	4	300	12%
jdd4 (24)	0.07	0.47	4	4	300	19%

TABLE 4.1 – Caractéristiques des données générées. Chaque ligne représente un jeu de données, le numéro entre parenthèses est l'identifiant de jeu de données et sera utilisé par la suite dans les données tabulaires. δ_1 et δ_2 sont les mesures respectives de suffisance et d'indépendance. $|\mathbf{X}_1|$ et $|\mathbf{X}_2|$ sont le nombre de colonnes des vues $\mathcal{X}_1$ et $\mathcal{X}_2$, N est le nombre d'échantillons dans l'ensemble de données et r_D est le ratio de déséquilibre, c'est-à-dire le pourcentage d'instances positives.

c'est-à-dire le pourcentage d'instances appartenant à la classe positive. Les indicateurs de suffisance et d'indépendance δ_1 et δ_2 de Ling et al [81] y sont également présents, estimés en utilisant un arbre de décision comme classificateur de base et une validation croisée. Selon l'étude de Ling et al [81], le co-training a tendance à améliorer les performances lorsque $\delta_1 < 0,23$ et $\delta_2 < 0,15$. Les quatre jeux de données générés respectent la première condition, mais aucun ne respecte la deuxième. Les vues ont donc une bonne suffisance mais ne sont pas vraiment indépendantes.

Adaptation des données UCI pour le co-training

Au total, cinq benchmarks du dépôt de référence UCI [33] ont été sélectionnés pour les expériences suivantes. Ces jeux de données ne sont pas originellement dédiés au co-training et les deux vues ont du être créées artificiellement. Pour ce faire, la procédure décrite par Ling et al [81] basée sur l'entropie pour diviser un jeu de données en deux vues, décrite dans la sous-section 2.3.2, a été utilisée. Les jeux de données UCI à deux vues sont partagés sur github [1] afin de créer une base de référence publique pour le co-training.

La table 4.2 reprend les mêmes critères que la table 4.1 mais pour les benchmarks UCI. Pour rappel, l'étude de Ling et al [81] qui dit que le co-training a tendance à améliorer les performances si $\delta_1 < 0,23$ et $\delta_2 < 0,15$. En comparant ces valeurs avec celles de la table 4.1, seuls les jeux de données 8 et 13 ont une bonne suffisance de leurs vues (respectivement $\delta_1 = 0,11$ et $\delta_1 = 0,19$), et seul le jeu de données 5 a une indépendance

1. https ://github.com/MathildeMonvoisin/Co-training-benchmark

	δ_1	δ_2	$\lvert\mathbf{X}_1\rvert$	$\lvert\mathbf{X}_2\rvert$	N	r_D
heart-statlog (5)	0.33	0.12	7	6	270	45%
APS failure (8)	0.11	0.39	5	5	758	36%
hydraulic stable (10)	0.45	0.66	8	7	2206	34%
hydraulic valve (12)	0.56	0.67	8	7	2206	67%
hydraulic leakage (13)	0.19	0.67	8	7	2206	22%

TABLE 4.2 – Caractéristiques des benchmarks UCI. Chaque ligne représente un jeu de données, le numéro entre parenthèses est l'identifiant de jeu de données et sera utilisé par la suite dans les données tabulaires. δ_1 et δ_2 sont les mesures respectives de suffisance et d'indépendance. $|\mathbf{X}_1|$ et $|\mathbf{X}_2|$ sont le nombre de colonnes des vues $\mathcal{X}_1$ et $\mathcal{X}_2$, N est le nombre d'échantillons dans l'ensemble de données et r_D est le ratio de déséquilibre, c'est-à-dire le pourcentage d'instances positives.

des vues favorable ($\delta_2 = 0,12$).

4.2.2 Protocole expérimental

L'implémentation des différentes stratégies d'apprentissage proposées dans la section 4.1 a été réalisée avec la bibliothèque C++ dédiée aux modèles graphiques probabilistes PILGRIM développé au sein de l'équipe DUKe du LS2N, basée sur la bibliothèque ProBT [2]. Les modèles appris à l'aide des stratégies de co-training non-supervisé proposées dans ce chapitre sont des classifieurs bayésiens naïfs. Ils ont l'avantage d'être applicables pour tous les problèmes de classification et sont reconnus pour leur bonnes performances relativement à leur simplicité, comme détaillé dans la section 1.4. Chaque classifieur bayésien naïf est appris à l'aide de chacune des trois stratégies *split*, *recursive* et *progressive* avec cinq validations croisées à 2 échantillons (stratified 5x2-*fold cross validation*). Les performances des trois stratégies sont comparées avec les résultats d'un apprentissage indépendant de chaque modèle de manière non-supervisée. La significativité des résultats est vérifiée à l'aide d'un test de Student (stats.ttest_rel() de scipy [128]) avec un risque $\alpha = 1\%$.

Les paramètres sont appris avec l'algorithme EM avec un seuil d'arrêt de 0,0001 sur la variation de la vraisemblance et un nombre d'itérations de $n_{step} = 15$ (choisi pour garantir la convergence dans toutes nos expériences) pour la stratégie *recursive*.

2. https ://www.probayes.com/

	0.01	0.05	0.1
0.01			
0.05			
0.1			

	0.01	0.05	0.1
0.01			
0.05			
0.1			

TABLE 4.3 – Résultat du test de Student avec $\alpha = 0.05$ pour la comparaison des valeurs de $\mu = 0.01, 0.05$, ou 0.1 pour la précision équilibrée (table de gauche) et la sensibilité (table de droite). Vert et rouge signifient respectivement que la valeur en colonne est significativement meilleure ou pire que la valeur en ligne. Les cases grises sont les cases dont les valeurs sont non significatives.

De plus, le but de la stratégie *progressive* est d'ajouter les étiquettes au fur et à mesure de l'apprentissage selon la confiance du classifieur en les différentes instances. Le nombre d'annotations ajoutées à chaque itération dépend du taux μ. Pour déterminer la meilleure valeur de μ, une étude préliminaire a été effectuée pour comparer les valeurs de $\mu = 1\%, 5\%$ ou 10%. La table 4.3 montre les résultats du test de Student pour ces valeurs, et permet de conclure avec un risque de $\alpha = 5\%$ que le meilleur taux est $\mu = 1\%$. C'est donc ce paramètre qui est utilisé pour la suite des expériences.

Enfin, des a priori de Dirichlet sont utilisés sur la variable classe pour éviter une mauvaise convergence due aux données déséquilibrées. Une étude préliminaire a permis de conclure sur les α optimaux pour la variable classe : α_0 et α_1 étant respectivement le nombre d'observations virtuelles de CNF et de SC. N étant le nombre total d'instances, il a été déterminé que les *a priori* seraient assignés comme indiqué par les équations 4.1, simulant donc une proportion de 70% des cas comme étant en CNF.

$$\begin{cases} \alpha_0 = 0.7 \times N \\ \alpha_1 = 0.3 \times N \end{cases} \tag{4.1}$$

4.2.3 Résultats

Dans cette section, u désigne les résultats de l'apprentissage simple, s et r respectivement les résultats des stratégies *split* et *recursive*, et enfin la stratégie *progressive* a deux ensembles de résultats : les résultats de la stratégie *progressive* issus de la dernière itération (quand l'ensemble des étiquettes ont été ajoutées au vecteur de vérité) sont notés p_m. Les résultats issus de l'itération précédant le coude dans la confiance en les étiquettes ajoutées pour la stratégie *progressive* sont notés p_c et sont détaillés dans la table 4.6. Cette section

		vue A					vue B				
	Stratégie	**u**	**s**	**r**	$\mathbf{p}_m$	$\mathbf{p}_c$	**u**	**s**	**r**	$\mathbf{p}_m$	$\mathbf{p}_c$
u	**préc. éq.**		1	3	0	3		0	1	1	1
	sensibilité		1	4	4	4		0	3	3	2
s	**préc. éq.**	0		3	4	5	0		3	2	4
	sensibilité	0		6	1	5	0		2	4	4
r	**préc. éq.**	1	2		3	5	3	5		3	3
	sensibilité	1	4		1	3	1	4		5	8
$\mathbf{p}_m$	**préc. éq.**	1	6	5		3	3	6	5		6
	sensibilité	3	8	8		5	3	3	2		5
$\mathbf{p}_c$	**préc. éq.**	1	4	2	5		2	4	4	3	
	sensibilité	0	4	4	4		1	5	4	5	

TABLE 4.4 – Somme du nombre de jeux de données obtenant des résultats significatifs pour les vues A et B : en colonne la stratégie est significativement meilleure que celle qui est en ligne.

utilise le package python Kneed [3], qui implémente l'algorithme Kneedle décrit dans la section 1.4.2 pour détecter le coude dans l'évolution de la confiance.

La table 4.4 donne la somme du nombre de jeux de données pour lesquels les résultats sont significatifs lorsque l'on compare deux stratégies entre elles. Le chiffre dans une case d'une colonne c et d'une ligne l est à lire comme le nombre total de jeux de données pour lesquels la stratégie c est meilleure que la stratégie l, résultats des expériences commençant par les vues $\mathcal{X}_1$ et $\mathcal{X}_2$ additionnés.

Dans certains cas, les stratégies de co-training offrent de meilleurs résultats que les stratégies simples, et dans d'autres cas, l'inverse est vrai. Les cas présentant un bon fonctionnement des stratégies sont présentés dans la première sous-section. La deuxième sous-section détaille les cas où les stratégies de co-training non-supervisé n'ont pas amélioré les résultats ou les ont dégradés, et expliquent les raisons de ces échecs.

Réussite des stratégies

La table 4.5 montrent que la stratégie *recursive* et *progressive*$_m$ ont tendance globalement à améliorer les résultats pour six sur neuf des benchmarks (10, 12, 13, 22, 23 et 24) pour la vue $\mathcal{X}_1$. Pour la stratégie progressive, le fait d'arrêter l'apprentissage à l'itération avant la chute de confiance (p_c) donne une grande amélioration par rapport au fait d'aller

3. https ://github.com/arvkevi/kneed

Début vue A

	Stratégie	vue A s-u	vue A r-u	vue A p_m-u	vue A p_c-u	vue B s-u	vue B r-u	vue B p_m-u	vue B p_c-u
5	Δ préc. éq.			-9.7				11.2	
	Δ sensibilité					-13.2			
8	Δ préc. éq.				22.0		-23.4	-13.6	
	Δ sensibilité				42.9				
10	Δ préc. éq.				-9.6				9.8
	Δ sensibilité		-9.9	-6.9		25.4	39.7	54.2	
12	Δ préc. éq.				2.8			-3.4	2.9
	Δ sensibilité		36.1	15.8	57.1		8.3	8.8	-4.8
13	Δ préc. éq.	7.2	8.0				-3.7		
	Δ sensibilité	5.1	12.9	23.2	38.8			18.4	
21	Δ préc. éq.						19.1	-21.0	
	Δ sensibilité			-16.9					64.0
22	Δ préc. éq.		6.8				-10.6	-17.0	-15.6
	Δ sensibilité			39.3		54.4	52.6	65.9	41.1
23	Δ préc. éq.		12.2		11.5			23.4	
	Δ sensibilité		35.1	-32.7	22.9	25.0	-32.7		
24	Δ préc. éq.		-21.9					-18.2	15.6
	Δ sensibilité		55.6	32.8					

Début vue B

	Stratégie	s-u	r-u	p_m-u	p_c-u	s-u	r-u	p_m-u	p_c-u
5	Δ préc. éq.	-8.8		-2.2					
	Δ sensibilité	32.3							
8	Δ préc. éq.	-14.8					-23.4	-25.9	-5.2
	Δ sensibilité	27.6		31.1	14.1			-23.6	-14.8
10	Δ préc. éq.	6.3		9.8				13.5	
	Δ sensibilité	7.2	-9.9				39.7	38.3	
12	Δ préc. éq.	-4.6		-7.8	3.5				
	Δ sensibilité	29.1	36.1	20.2			8.3	7.4	55.6
13	Δ préc. éq.	7.6	8.0				-3.7	-3.6	-10.2
	Δ sensibilité	30.7	12.9						
21	Δ préc. éq.						19.1		
	Δ sensibilité			-42.4	-31.0				67.9
22	Δ préc. éq.	9.4	6.8				-10.6	-15.0	
	Δ sensibilité				-39.0		52.6	70.0	
23	Δ préc. éq.		12.2	32.6					17.0
	Δ sensibilité		35.1				-32.7	-34.8	
24	Δ préc. éq.	-10.4	-21.9		-18.6				
	Δ sensibilité		55.6		60.0			-29.4	

TABLE 4.5 – Différence de performances entre le co-training et l'apprentissage non-supervisé (u). Les cases vertes ou rouge correspondent à une différence positive (vert foncé pour le meilleur) ou négative. Les cases blanches sont les résultats non significatifs.

jusqu'à la fin pour ajouter toutes les étiquettes (p_m) pour trois benchmarks (8, 12 et 13), et améliore particulièrement la sensibilité de quatre benchmarks (21, 22, 23, 24).

En comptabilisant les scores significatifs dans la table 4.5, les stratégies *split*, *recursive* et *progressive* augmentent les performances d'un nombre de jeux de données en moyenne de 4,5 de 5,5 de 8 (p_m) et de 5,5 (p_c). La stratégie *progressive* obtient la meilleure moyenne en augmentation de performances, et *split* obtient le moins d'échecs mais a de manière générale peu de résultats significatifs. De plus, comme vu précédemment, p_m et p_c se complètent bien, car utiliser p_c lorsque la confiance diminue en fin d'apprentissage permet de corriger une partie des dégradations de performances observées pour p_m. Enfin, la table 4.6 souligne les différences entre p_c et p_m, et permet de conclure que p_c offre une grande amélioration de la sensibilité. Il est donc important de noter que c'est $progressive_m$ qui offre les meilleurs améliorations, au prix de grosses dégradations dans certains cas, mais que ces cas sont détectables et corrigeables grâce à $progressive_c$.

La table 4.7 expose les durées d'entraînement pour chaque jeu de données et chaque stratégie. Pour les jeux de données contenant le moins d'instances, les durées varient faiblement d'une stratégie à l'autre. Par exemple, 5 (270 instances) ne prend jamais plus de dix secondes pour effectuer un entraînement. En revanche, la durée de l'entraînement augmente de façon exponentielle avec le nombre de lignes. Le benchmark 8 a 750 lignes et met une vingtaine de secondes à tourner pour la stratégie progressive, tandis que les benchmarks 10, 12 et 13 ont 2200 lignes et mettent entre 80 et 90 secondes pour la totalité de l'entraînement progressive.

Si l'on cherche un entraînement rapide ou que l'on a beaucoup de données, *recursive* sera certainement un bon compromis entre un temps de calcul raisonnable et des résultats intéressants. Si l'on s'intéresse uniquement aux performances, $progressive_m$ est meilleure dans une grande partie des cas (ceux où la confiance des classifieurs est grande) et $progressive_c$ permet notamment de corriger certaines boucles de rétroaction négative (ceux où la confiance des classifieurs est faible). De plus, la stratégie *progressive* permet généralement d'augmenter significativement la sensibilité.

Echec des stratégies

On peut considérer que les stratégies de co-training ont échouées lorsque l'apprentissage simple (u) donne des résultats significativement meilleurs que les stratégies de co-training proposées (r,s,p).

Le premier cas de figure de l'échec des stratégies de co-training non-supervisées est

		Début vue A		Début vue B	
	Stratégie	**vue A** p_c-p_m	**vue B** p_c-p_m	**vue A** p_c-p_m	**vue B** p_c-p_m
5	**Δ préc. éq.**	-0.8	-14.0	-4.5	3.7
	Δ sensibilité	2.7	-22.1	11.0	0.7
8	**Δ préc. éq.**	0.1	5.5	-10.2	15.9
	Δ sensibilité	47.0	7.0	-28.0	15.6
10	**Δ préc. éq.**	1.5	-0.9	-10.1	-15.5
	Δ sensibilité	-15.9	-33.6	-11.4	-7.0
12	**Δ préc. éq.**	2.5	5.6	5.0	-1.4
	Δ sensibilité	-4.7	-5.5	43.1	-21.6
13	**Δ préc. éq.**	-4.5	-6.3	17.0	1.6
	Δ sensibilité	7.5	-15.7	48.8	-16.9
21	**Δ préc. éq.**	-23.7	31.4	-23.9	0.8
	Δ sensibilité	28.4	21.2	46.4	56.2
22	**Δ préc. éq.**	10.0	11.3	-17.3	2.8
	Δ sensibilité	-78.5	-26.5	54.5	-46.2
23	**Δ préc. éq.**	-2.5	-10.7	-8.9	4.3
	Δ sensibilité	33.2	-21.5	-20.0	7.0
24	**Δ préc. éq.**	-3.0	11.1	-8.4	-22.0
	Δ sensibilité	-7.6	-20.0	31.4	2.8

TABLE 4.6 – Comparaison des indicateurs de précision équilibrée et de sensibilité pour *progressive*, à la dernière itération (p_m) et à l'itération du coude (p_c). Les cases sont vertes lorsque la différence entre les stratégies étudiées est positive, et rouge quand elle est négative. La couleur verte ou rouge est plus foncée lorsque la valeur absolue de cette différence est supérieure à 20. Les cases grises sont les résultats non significatifs.

celui des jeux de données qui ne sont pas adaptés au co-training. Dans la table 4.4, on remarque que les stratégies, *recursive* et *progressive* dégradent les performances en précision équilibrée d'un des jeux de données, et de plusieurs d'entre eux en sensibilité, notamment pour les résultats p_m. La table 4.8 montre le détail des jeux de données pour lesquels chaque stratégie de co-training échoue par rapport à l'apprentissage simple. Ce sont les jeux de données 8, 10 et 13, qui apparaissent le plus dans cette table, on peut donc considérer que les stratégies de co-training échouent pour ces benchmarks, notamment *split* et *recursive*. Les indicateurs d'indépendance de ces jeux de données sont particulièrement élevés : respectivement 0.39, 0.66 et 0.57, c'est-à-dire bien au-delà des valeurs avancées par Ling et al [81] à partir desquelles le co-training a tendance à dégrader les performances ($\delta_2 > 0.26$). L'échec des stratégies proposées *split* et *recursive* sur les jeux de données 8, 10 et 13 peut donc s'expliquer par le fait qu'ils soient intrinsèquement inadaptés au

	u	s	r	p_m	p_c
5	0.2	0.3	1.2	9.9	8.1
8	1.0	1.4	3.5	21.4	16.6
10	10.0	14.8	23.4	83.4	80.7
12	13.9	18.2	26.8	82.7	80.1
13	13.5	18.5	28.3	90.2	87.2
21	0.1	0.2	1.0	8.1	6.4
22	0.2	0.3	1.1	8.1	6.1
23	0.2	0.3	1.0	8.2	5.7
24	0.1	0.2	0.9	8.1	4.4

TABLE 4.7 – Durée moyenne (en secondes) de l'entraînement pour chaque jeu de données et chaque stratégie.

Indicateur	**Précision équilibrée**				**Sensibilité**			
Stratégie	**s**	**r**	p_m	p_c	**s**	**r**	p_m	p_c
A		24	5	10		10	10, 21, 23	
B		8, 13, 22	8, 13, 22	8, 13		23	8, 23, 24	8

TABLE 4.8 – Jeux de données pour lesquels les stratégies de co-training (en colonne) dégradent les performances par rapport à l'apprentissage simple (détail des colonnes u de la table 4.4).

co-training.

Le deuxième cas pouvant favoriser de mauvaises performances, notamment par la stratégie *progressive*, est une chute de la confiance du modèle en les nouvelles instances qu'il doit classifier. La table 4.8 montre que les jeux de données 22 et 24 donnent de moins bons résultats pour $progressive_m$ que pour l'apprentissage simple, notamment sur la vue $\mathcal{X}_2$. La figure 4.2 montre la confiance dans le dernier étiquetage effectué en fonction du critère d'arrêt, et il est visible que la confiance est nettement plus basse pour les jeux de données 8 et 21 à 24 sur le graphe des ajouts de p_m. Cette observation porte en la faveur que les jeux de données 22 et 24 entrent dans une boucle de rétroaction négative à partir d'une certaine itération, ultérieure à l'itération du coude détecté. La table 4.2 de p_c montre qu'avant leur chute de confiance, ces deux jeux de données ont une confiance aussi élevée que les autres (probabilité aux alentours de 0.8 pour la vue $\mathcal{X}_1$, et qui ne dépasse pas 0.75 pour la vue $\mathcal{X}_2$). De plus, 22 et 24 n'apparaissent pas dans la colonne p_c de la table 4.8, ce qui signifie que la boucle de rétroaction négative a été contrée par la stratégie qui consiste à arrêter l'entraînement avant la chute de confiance (p_c). Le même phénomène est observables pour le benchmark 21, qui a des performances

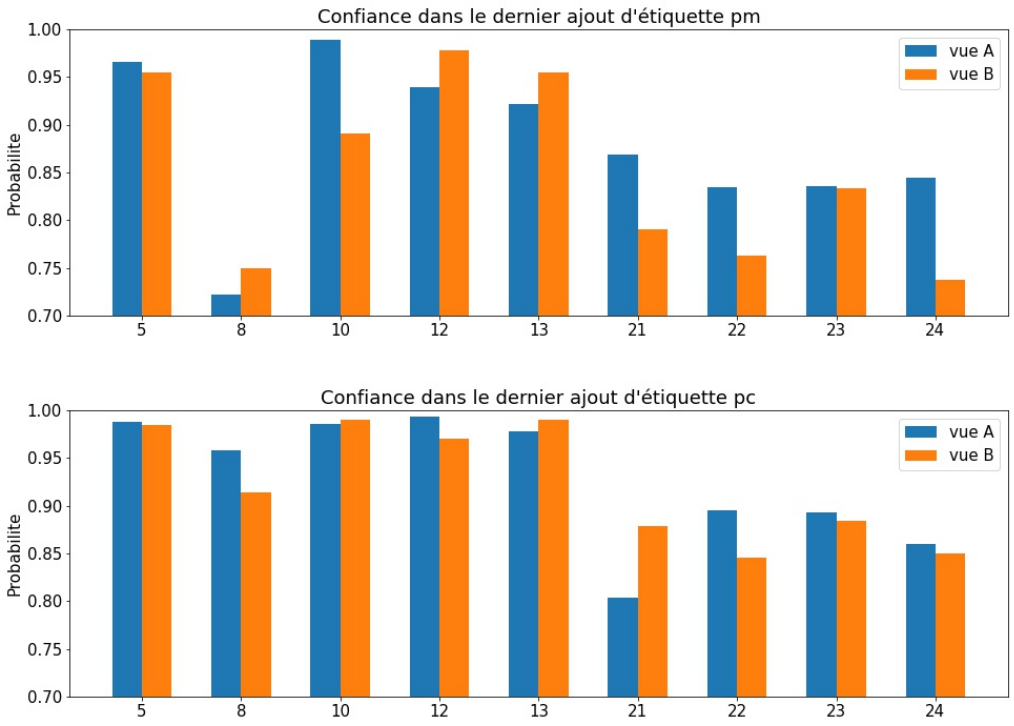


FIGURE 4.2 – Confiance dans les ajouts d'étiquettes à l'itération des résultats p_m et p_c pour chaque jeu de données. Moyenne sur début A et début B.

dégradées pour p_m. Une confiance faible du classifieur dans les étiquettes utilisées pour la prévision apparaît comme pouvant dégrader grandement les performances du co-training non-supervisé.

Les observations ci-dessus permettent donc de suggérer que deux cas de figures entraînent l'échec des stratégies de co-training : la première est le non-respect des hypothèses intrinsèques au co-training, bien qu'ici le co-training améliore les performances à des seuils au-delà de ceux avancés par l'étude de Ling et al [81]. La deuxième cause de dégradations de performances est une baisse de confiance dans les prédictions utilisées pour la construction du vecteur de vérité. Ce deuxième cas concerne plutôt les stratégies *recursive* et $progressive_m$.

4.3 Conclusion

Ce chapitre a présenté trois stratégies de co-training non-supervisées, qui peuvent être appliquées à tout classifieur probabiliste et divers domaines d'application.

Ces stratégies ont été illustrées à l'aide de plusieurs benchmarks disponibles dans le dépôt UCI (et adaptés à un objectif de co-training) et de jeux de données artificiels. La section 4.2 a montré que ces stratégies peuvent tirer profit d'informations distinctes afin de fournir de meilleurs résultats, pour des durées d'entraînement raisonnables.

Les observations précédentes conduisent à penser qu'il existe un grand bénéfice pour les performances à arrêter la stratégie *progressive* avant qu'elle ait ajouté toutes les étiquettes au vecteur de prédiction (p_c), au moins pour les jeux de données qui atteignent une confiance très faible à la fin des ajouts. *progressive*$_c$ permet souvent d'augmenter beaucoup la sensibilité, ce qui peut être un atout pour certaines applications où manquer une classification positive peut avoir des conséquences graves. A l'inverse, de manière générale, *recursive* a plus tendance à permettre de grandes améliorations de précision équilibrée, et *progressive*$_m$ offre des bonnes améliorations de manière plus globale. Enfin, il est important de souligner que ces expériences ont été effectuées avec le classifieur bayésien le plus simple, le classifieur bayésien naïf, et il n'est donc pas exclu que les stratégies de co-training non-supervisé proposées dans ce chapitre permettent de plus grandes améliorations avec des classifieurs plus adaptés au problème.

Ling et al [81] énoncent dans leurs conclusions que le co-training a tendance à fonctionner sur des jeux de données quand $\delta_1 < 0.23$ et $\delta_2 < 0.15$. Aucun des jeux de données utilisés dans ce chapitre ne rentre dans ces intervalles, tous les jeux de données ont pourtant reçu une amélioration avec au moins une des stratégies de co-training non-supervisé.

Ce travail peut encore être étendu ou amélioré de plusieurs façons. Premièrement, comme mentionné précédemment, le choix a été d'utiliser un classifieur naïf bayésien, mais il serait également intéressant de reproduire les expériences avec d'autres classifieurs connus, tel que le TAN, le BAN, etc, pour voir si les mêmes conclusions ressortent.

La stratégie *progressive* avec le critère d'arrêt du coude permet des améliorations mais dégrade les performances relevées pour certains jeux de données. Il est possible que ces baisses de précision et sensibilité soient dues à un critère d'arrêt inadapté. Il est possible qu'un autre critère que la recherche d'un coude dans la confiance d'ajout dans les étiquettes soit plus pertinent, par exemple un seuil de confiance sous lequel ne pas tomber. Une recherche plus développée des conditions d'arrêt de la stratégie *progressive* pourrait venir compléter ce travail sur les stratégies de co-training. La stratégie *recursive* pourrait également bénéficier d'un critère d'arrêt différent : au lieu de contrôler le nombre d'étapes, il serait possible de surveiller la vraisemblance et arrêter les itérations lorsque celle-ci a cessé de s'améliorer de manière significative.

La boucle de rétroaction négative observée lors de certaines expériences est partiellement corrigée par la stratégie progressive, mais il serait possible de raffiner plus encore le passage d'information en ne transférant pas l'étiquette de la classe la plus certaine, mais la distribution de probabilité de l'état comme une *soft evidence*, comme proposé dans le contexte multi-agent par Vomlel [129].

Les hypothèses de suffisance ne sont également pas toujours respectées dans les applications réelles, il serait possible d'étendre ce travail sur le co-training non-supervisé avec des vues insuffisantes, comme proposé par Guo et al [46].

En résumé

- ✓ Les stratégies de co-training non-supervisé proposées ont été montrées comme performantes sur plusieurs benchmarks annotés, notamment *recursive* et $progressive_m$.
- ✓ Lorsque la confiance du classifieur diminue en dessous d'un certain seuil, le critère d'arrêt alternatif $progressive_c$ permet d'améliorer les résultats par rapport à $progressive_m$, et notamment la sensibilité.
- ✓ La stratégie *progressive* permet généralement d'augmenter significativement la sensibilité.
- ✓ Les stratégies proposées prouvent leur utilité même en dehors du cas idéal d'une indépendance et d'une suffisance élevée des jeux de données.
- ✓ La généricité des algorithmes les rend directement utilisables pour tout classifieur probabiliste.
- ✓ La durée d'entraînement est raisonnable (jusqu'à une centaine de secondes) pour un jeu de données jusqu'à 2200 lignes et 7 colonnes (non testée au-delà).

Chapitre 5

Co-training de Réseaux bayésiens pour le diagnostic en usinage

L'usinage est l'un des principaux procédés de fabrication mécanique, en particulier pour les pièces à forte valeur ajoutée. Il est effectué sur des machines-outils, qui sont très performantes et dynamiques. Leur productivité est notamment liée aux performances de la broche. Elle fait tourner l'outil, permettant le mouvement de coupe qui retire la matière. En usinage à grande vitesse (UGV), elles tournent très vite (ex. N = 30 000 tr/min), ce qui permet d'enlever des volumes de matière plus rapidement. Étant très performantes et aux limites des possibilités technologiques, les broches sont malheureusement également fragiles et sont ainsi le talon d'Achille des machines-outils d'UGV. La figure 5.1 montre une photographie de broche et de machine outil.

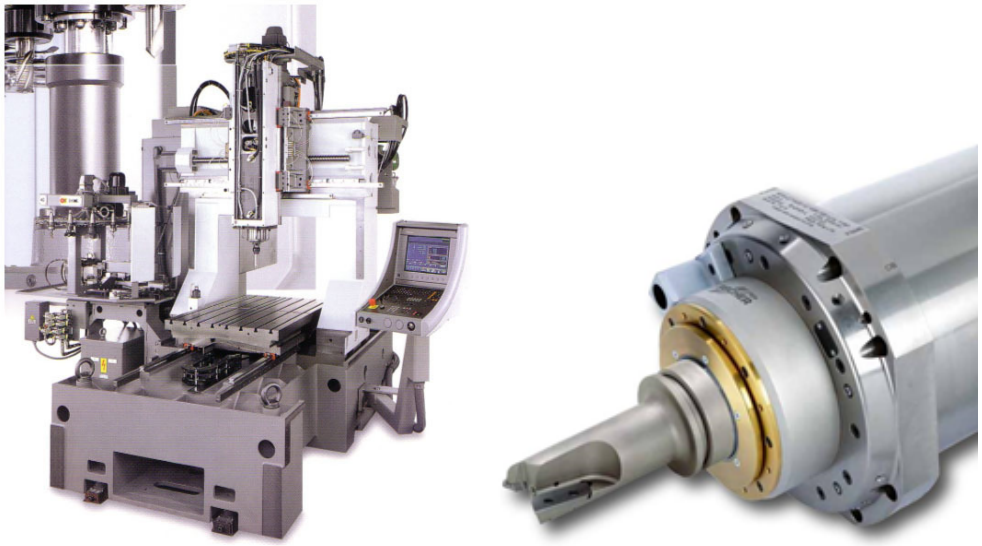


FIGURE 5.1 – Photographie d'une machine outil (à gauche) et d'une broche (à droite).

Ces broches ont des durées de vies très variables d'une machine à l'autre, souvent beaucoup plus courtes qu'elles ne devraient l'être par rapport aux spécifications des constructeurs. Les experts du domaine connaissent certains types d'événements qui pourraient engendrer cette défaillance prématurée, mais peuvent rarement savoir pour une broche donnée quelles sont les facteurs qui ont été les plus impactants pour la rendre hors d'usage. En effet, les dégradation des broches sont lentes et peuvent durer quelques mois à plusieurs années. Elles sont donc difficilement perceptibles par l'homme et difficiles à étudier scientifiquement au cours de réelles productions industrielles, qui elles-seules pourront générer des défaillances réalistes. Une instrumentation vibratoire pointue est nécessaire pour pouvoir suivre l'évolution de l'état de la broche, tel que cela se fait en maintenance conditionnelle.

L'identification des raisons qui créent les différents types de dégradations est un sujet capital pour le domaine de l'UGV, car bien que les problèmes graves y soient très rares, le fait que les broches aient des durées de vie aussi variables pour des raisons incomprises provoque de coûteux arrêts de production et de maintenance curative. Il est donc crucial de déterminer quel type et quelle intensité d'incident en usinage va endommager les broches, d'une part ; et quelles évolutions de l'état de la broche sont significatives, d'autre part.

L'application qui motive ce chapitre est un jeu de données collecté dans un contexte de production industrielle d'usinage à grande vitesse, avec une partie du jeu de données issue de la surveillance du procédé d'usinage, et l'autre partie de la signature vibratoire

de la broche. L'ensemble des indicateurs et critères utilisés pour effectuer le suivi de l'électrobroche en UGV et détecter les incidents qui peuvent survenir en usinage sont détaillés section 5.1. La section 5.2 décrit la méthode de récolte des données ainsi que les caractéristiques du jeu de données. Enfin, la section 5.3 détaille la compatibilité de ces deux parties du jeux de données avec les stratégies de co-training non-supervisé introduites dans le chapitre 4, notamment par le fait que les deux parties du jeu de données permettent de détecter les endommagements de la broche, et peuvent apprises ensemble puis être utilisées séparément pour effectuer les inférences.

5.1 Éléments pour le diagnostic en usinage

Il est possible d'évaluer l'état de la broche à un moment donné en procédant à une signature vibratoire de la broche. C'est une procédure répandue dans le domaine industriel et bien connue par les experts en maintenance conditionnelle [108]. De Castelbajac et al [27] ont proposé de surveiller en outre le procédé d'usinage, ce qui peut également donner des indications sur ce que la machine a subi pendant une période de temps donnée, et donc aider à savoir comment la broche a été endommagée pendant cette période.

Les différents types d'événements qui peuvent se produire sont notamment le broutement, les bris d'outils ou les chocs [41].

5.1.1 Signature vibratoire de l'état de la broche

Une signature vibratoire de broche permet d'obtenir le spectre fréquentiel des vibrations, qui sont mesurées en faisant tourner un outil à vide, toujours à iso-conditions. Le niveau vibratoire de la machine donne des indications sur l'évolution de son état, cf [norme ISO17243]. Pour des analyses plus fines, l'analyse du spectre fréquentiel permet de détecter des défauts d'un des composants d'un système mécanique. En particulier, les fréquences caractéristiques des défauts d'un roulement permet de suivre sa dégradation, au travers de la fréquence de passage des billes sur la piste extérieure et intérieure, respectivement désignées $BPFO$ (*Ball Pass Frequency of the Outer Race*) ou $BPFI$ (*Ball Pass Frequency of the Inner Race*), donnent une indication sur l'état des pistes extérieures et intérieures des bagues. La fréquence fondamentale du train FTF détecte les défauts de la cage et BSF les défauts des billes. L'équilibrage de l'arbre $1X$ peut être suivi au travers d'une estimation de son balourd, par l'amplitude de la contribution à la fréquence de

rotation du rotor, notée 1X [41].

5.1.2 Surveillance du procédé

La surveillance du procédé d'usinage consiste à analyser les signaux enregistrées pendant l'usinage, afin de détecter de potentiels incidents, tels que les bris d'outils ou de prédire la qualité de la surface de la pièce usinée. Une variété de critères existent pour ce faire, les plus fréquents étant les niveaux vibratoires globaux $A_{rms}(t)$ et $V_{rms}(t)$, respectivement en accélération ($m.s^{-2}$) et en vitesse ($mm.s^{-1}$). $V_{rms}(t)$ et $A_{rms}(t)$ sont respectivement la valeur efficace de la vitesse et de l'accélération de la vibration, révélant le niveau de vibration à chaque instant.

Godreau et al [41] ont également proposé deux nouveaux critères de surveillance : $N_h(t)$ et $U_b(t)$. Le critère de surveillance $N_h(t)$ permet la détection du broutement. Le broutement est un phénomène de coupe instable qui entraîne un défaut de qualité inacceptable (des ondulations) sur la surface de la pièce [92]. L'analyse du spectre de vibration révèle que, en présence de broutement, de nouvelles contributions apparaissent à des fréquences non-harmoniques (de la vitesse de la broche). $N_h(t)$ est la somme des amplitudes des cinq contributions non-harmoniques dominantes du spectre de vibration. Le critère $U_b(t)$ quant à lui, détecte les bris d'outil, qui peuvent également se produire pendant l'usinage. Un bris d'outil conduit à une augmentation du déséquilibre mécanique de l'outil, qui peut être estimé par l'amplitude des vibrations à la fréquence de la broche. $U_b(t)$ a donc été défini par Godreau et al [41] comme l'amplitude de la contribution à la fréquence de rotation de la broche, obtenue dans le spectre des vibrations en effectuant un suivi d'ordre (*order tracking*). Le critère $U_b(t)$ est évalué lorsque la broche est en rotation mais pas en train d'usiner, pour éviter les perturbations dues aux efforts de coupe. Cette surveillance polyvalente permet de détecter de manière fiable les défaillances de l'outil pendant l'exploration des données.

5.2 Description des données UGV

Les données d'UGV ont été collectées dans un contexte réel de production industrielle aéronautique de pièces de structure en alliage d'aluminium. Elles portent sur trois durées de vie complètes de broche, collectées sur un total 436 jours de données agrégées et non étiquetées [41].

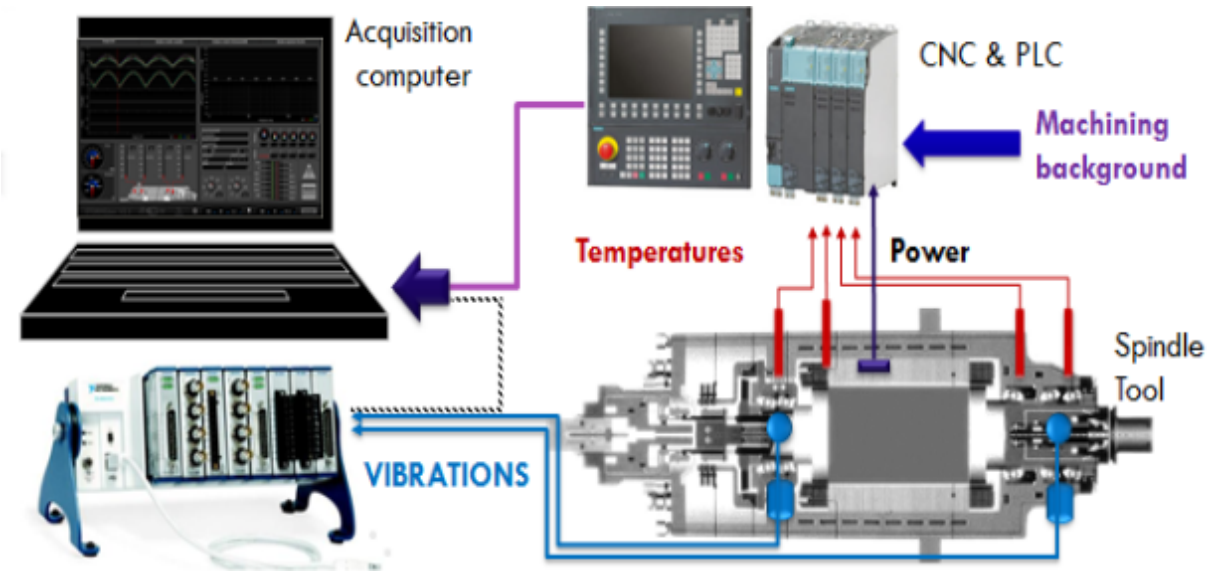


FIGURE 5.2 – Système EmmaTools de collecte des données d'UGV. Illustration issue des travaux de Godreau et al [41].

5.2.1 Données de surveillance de l'état de la broche

La surveillance de l'état de la broche se fait à partir des données de signature vibratoire de la broche. Ces dernières ont été enregistrées quotidiennement à 25.6kHz lors d'une rotation à une vitesse de 17 500 tours par minute. Cette prise de mesures a été réalisée après avoir fait chauffer la broche, pour qu'elle soit toujours à la même température à chaque enregistrement des données.

Les composantes contenues dans la vue B, issues de la signature de broche, sont les critères $BPFO$, $BPFI$ et FTF pour les défauts de roulement, ainsi que l'équilibrage $1X$ pour les rotors déséquilibrés ou voilés.

5.2.2 Données de surveillance du procédé d'usinage

Les quatre critères présentés dans la section 5.1.2 sont présents dans les données de surveillance de procédé du jeu de données d'UGV. Ces critères bruts de surveillance ($A_{rms}(t)$, $V_{rms}(t)$, $N_h(t)$ et $U_b(t)$) sont obtenus par traitement du signal en temps réel à partir de quatre accéléromètres qui fournissent des signaux vibratoires de 25kHz et ils sont collectés à une fréquence de 10 Hz, ce qui correspond à 864000 valeurs par jour. Ces valeurs sont collectées et enregistrés grâce au système EmmaTools illustré figure 5.2, qui a été développé par le LS2N.

Pour être comparés à l'évaluation quotidienne de l'état de la broche, les critères de surveillance du procédé doivent être agrégés quotidiennement. Les incidents survenant pendant l'usinage pouvant être très brefs, ils ne peuvent pas être détectés en faisant la moyenne de toutes les valeurs de surveillance mesurées pendant une journée. En outre, seules des vibrations importantes, dépassant un certain seuil, peuvent endommager la broche. La proposition d'agrégation de Castebelbajac et al [27] d'agréger ces critères bruts de surveillance d'usinage en fonction de différents seuils τ comme décrit dans l'équation 3.1 (définie section 3.1), est donc reprise pour cette étude. Des valeurs élevées de τ révèlent des événements courts et violents, alors que des valeurs faibles mettent en évidence des événements longs et d'intensité modérée.

Chaque critère brut est agrégé pour 10 valeurs différentes de seuil τ, résultant chacun en 10 critères avancés : $\tau_{A_{rms}} = \tau_{N_h} = \{20, 40, 60, ..., 180, 200\}$ en $m.s^{-2}$ et $\tau_{V_{rms}} = \tau_{U_b} = \{2, 4, 6, ..., 18, 20\}$ en $mm.s^{-1}$ pour V_{rms} et $m.s^{-2}$ pour U_b. Une analyse de Godreau et al [41] par algorithme génétique a donné les critères optimaux A^{120}_{rms}, V^{4}_{rms}, N^{100}_{h} et U^{10}_{b} avec les seuils optimaux $\tau_{A_{rms}} = 120$, $\tau_{V_{rms}} = 4$, $\tau_{N_h} = 100$ et $\tau_{U_b} = 10$.

5.2.3 Vérité

La vérité pour ce jeu de données est issue d'une analyse de données réalisée par Godreau et al [41]. C'est un vecteur binaire $\mathbf{y}$ qui indique s'il y a eu une dégradation ($y = 1$) ou si tout s'est déroulé sans accroche ($y = 0$) un jour donné. A partir des ensembles de données respectivement de surveillance du procédé et de surveillance de l'état de la broche, seuls 6 (1,6 %) et 4 (1,1 %) événements ont été identifiés dans cette étude comme des événements qui auraient pu sérieusement endommager l'une des trois broches suivies.

Des experts en UGV ont par la suite confirmé les dégradations trouvées. Cependant, peu de journées annotées comme sans événements majeurs ont été vérifiées. D'autres évènements ont pu survenir et être restés non détectés à ce stade.

5.3 Stratégie d'apprentissage

Pour le diagnostic de l'électrobroche, les modèles utilisés sont présentés dans la sous-section 5.3.1. La stratégie d'apprentissage des paramètres est présentée dans la sous-section 5.3.2.

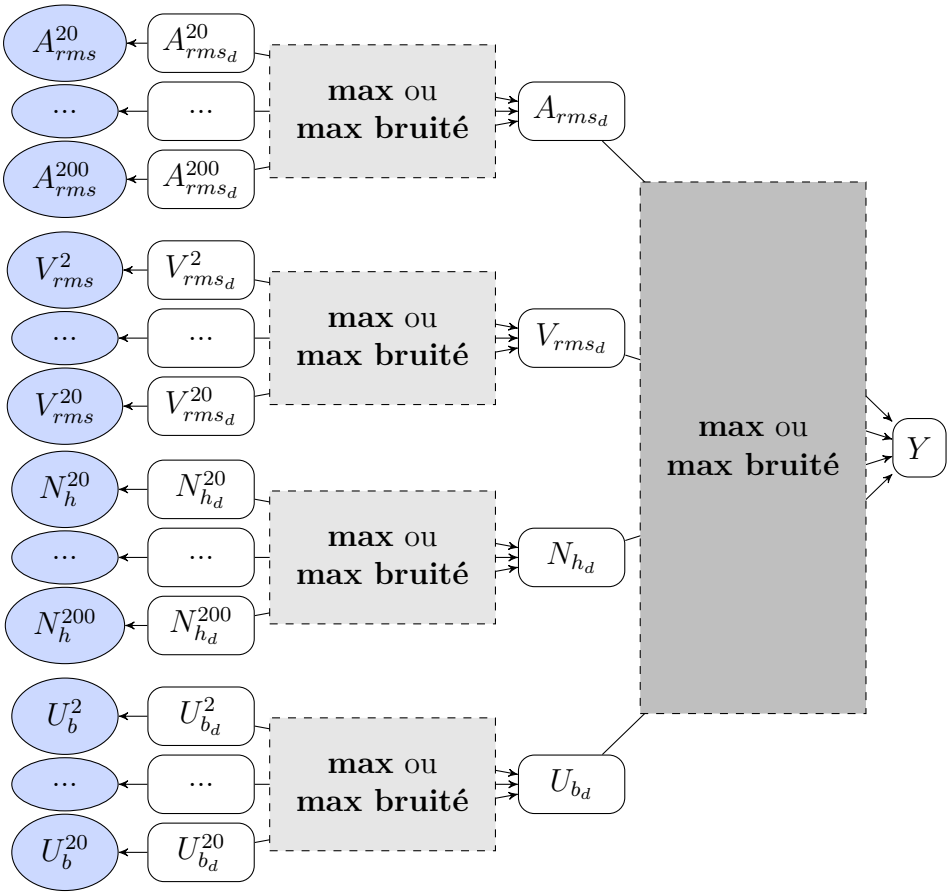


FIGURE 5.3 – Architecture du modèle de la surveillance du procédé d'usinage.

5.3.1 Modèle proposé

Les entrées du modèle sont les critères de surveillance avancés détaillés dans la section 5.2. Lorsque plusieurs seuils τ sont utilisés pour chaque critère de surveillance, la structure du modèle A est celle qui se situe dans la figure 5.3. Il s'agit d'une application de la structure générique présentée dans le chapitre 3 (figure 3.3). Si seul le seuil considéré comme optimal est utilisé en entrée du modèle, alors l'agrégation locale n'est pas pertinente et la structure se simplifie telle que sur la figure 5.4.

Le modèle B a pour variables d'entrée $\Delta(BPFO)$, $\Delta(BPFI)$, $\Delta(FTF)$ et $\Delta(1X)$, sans problématique de seuil. Le fait d'utiliser comme entrée $\Delta(X_i)$ pour chaque critère X_i, c'est-à-dire la valeur d'un jour donné à laquelle la valeur de la veille est soustraite, permet de détecter des évolutions soudaines de l'état de la broche. C'est également une version simplifiée de la structure du chapitre 3 (figure 3.3) qui est utilisée pour ce modèle. Elle est représentée figure 5.5.

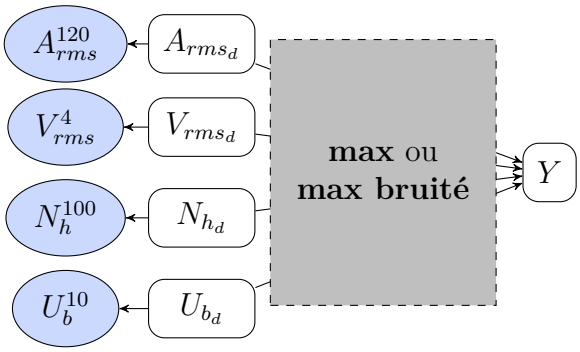


FIGURE 5.4 – Architecture simplifiée du modèle de surveillance du procédé d'usinage lors de l'utilisation des seuils optimaux.

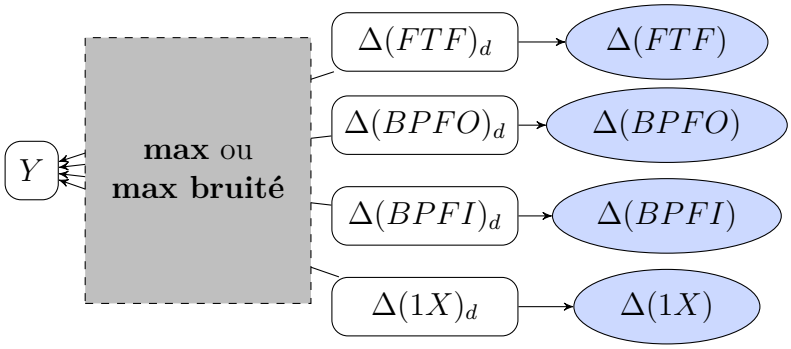


FIGURE 5.5 – Architecture du modèle de signature vibratoire de la broche.

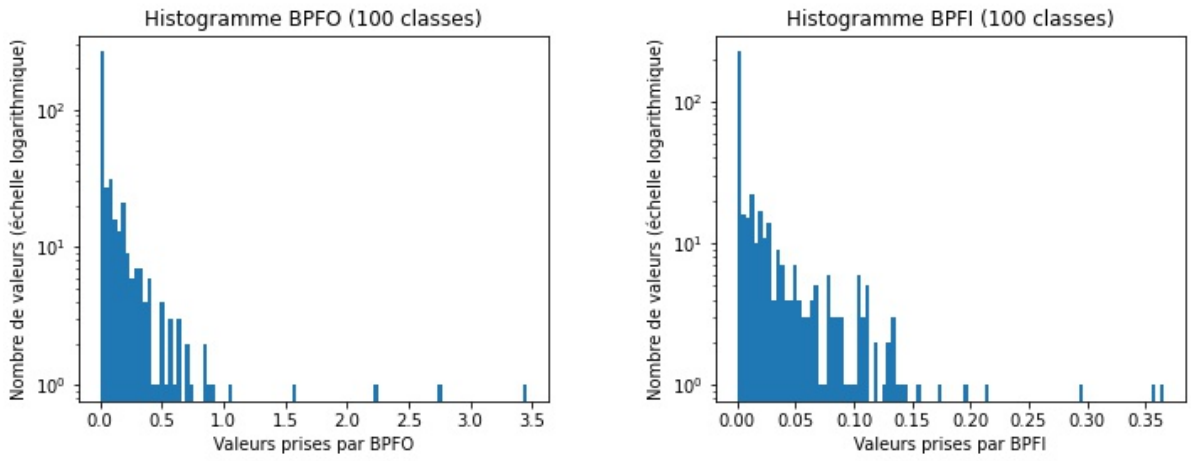


FIGURE 5.6 – Distribution des valeurs de $\Delta(BPFO)$ et $\Delta(BPFI)$ (données B de signature vibratoire de la broche). Échelle logarithmique sur l'axe des ordonnées.

5.3.2 Apprentissage des paramètres

Une approche classique de cette problématique de maintenance conditionnelle serait d'apprendre le modèle d'état de la broche de manière supervisée, avec la vérité du terrain de la dégradation quotidienne de la machine étudiée. Cependant, dans le cadre d'une application industrielle sans données annotées, il est impossible d'apprendre le modèle de manière supervisée. Ici, le jeu de données est naturellement divisé en deux parties qui donnent toutes les deux des informations différentes sur l'état de la broche à un instant donné. C'est un contexte très favorable au co-training.

Par conséquent, pour les expériences à suivre, les deux réseaux bayésiens construits à partir du modèle présenté dans le chapitre 3 pour modéliser respectivement la surveillance de procédé (modèle A) et la surveillance de l'état de la broche (modèle B) sont appris à l'aide des stratégies de co-training non-supervisé présentées dans le chapitre 4.

Toutes les variables des modèles A et B sont discrétisées à l'aide de trois intervalles qui correspondent à des niveaux de gravité {OK, dégradé, KO}. Ce choix a été fait à partir de l'observation de la répartition des données, confirmé par un avis expert et par une étude détaillée sur le sujet [58]. La distribution des données A (seuils optimaux) et B est visible sur les histogrammes des figures 5.6, 5.7, 5.8 et 5.9.

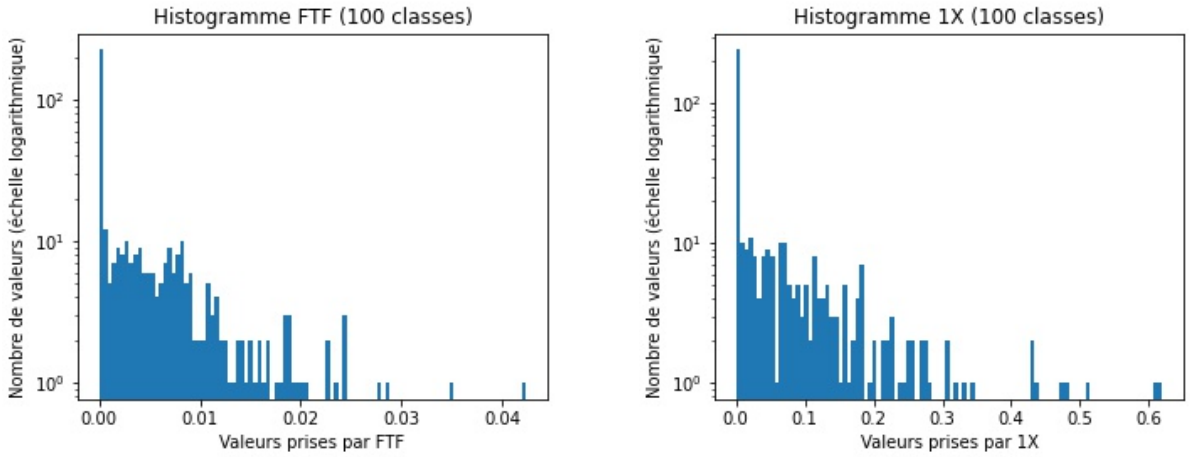


FIGURE 5.7 – Distribution des valeurs de $\Delta(FTF)$ et $\Delta(1X)$ (données B de signature vibratoire de la broche). Échelle logarithmique sur l'axe des ordonnées.

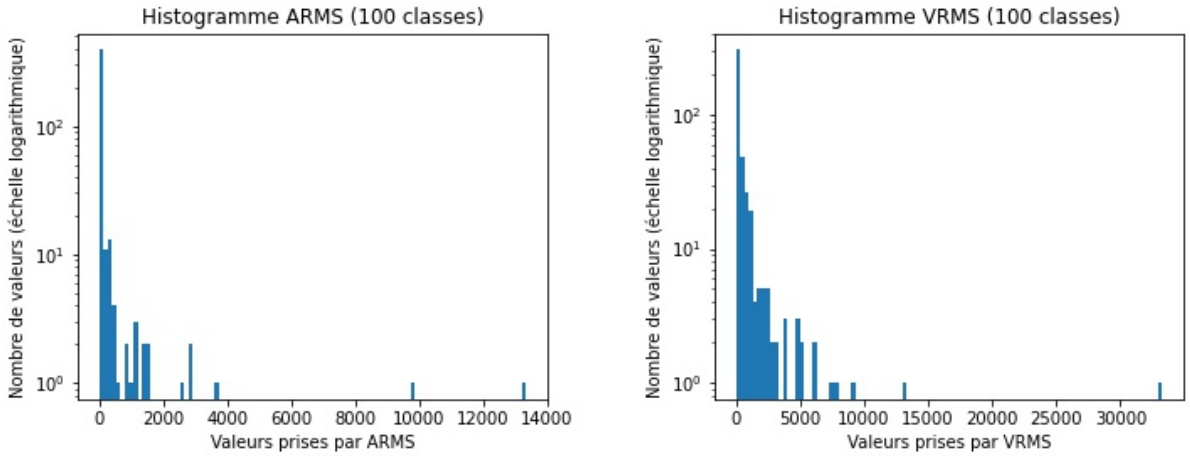


FIGURE 5.8 – Distribution des valeurs de A^{120}_{rms} et V^{4}_{rms} pour les seuils critiques respectivement $\tau_{A_{rms}} = 120$ et $\tau_{V_{rms}} = 4$. Échelle logarithmique sur l'axe des ordonnées.

5.4 Expériences sur données UGV

Les objectifs des expériences suivantes sont multiples : tout d'abord, évaluer les performances en prédiction des stratégies de co-training appliquées sur la structure présentée. Ensuite, analyser les contributions des différents seuils pour les comparer aux conclusions tirées sur les seuils optimaux par Godreau et al [41], et montrer l'intérêt de l'approche de co-training proposée en l'appliquant à des données réelles de fabrication. Dans ce but, un large ensemble d'expériences a été mené sur le jeu de données d'usinage présenté section

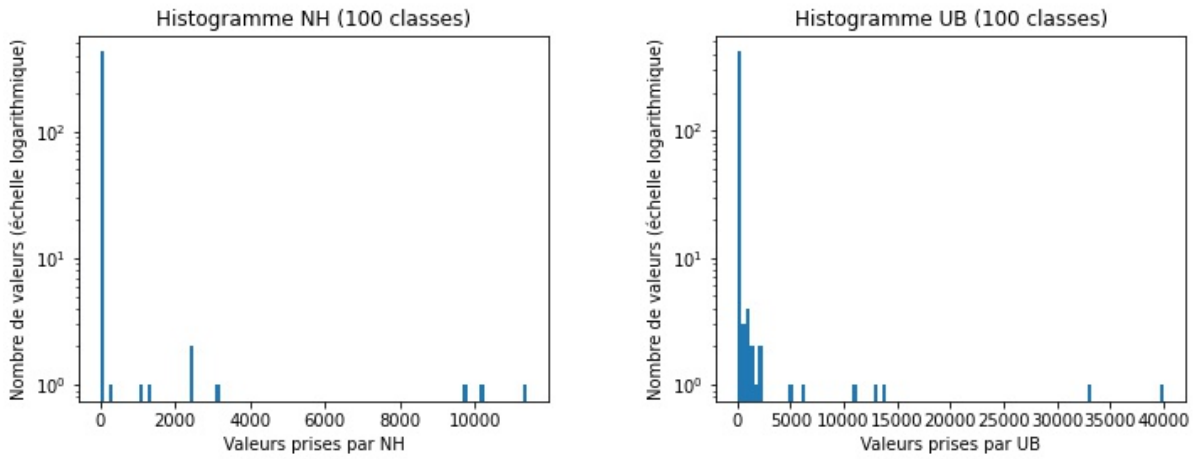


FIGURE 5.9 – Distribution des valeurs de N_h^{100} et U_b^{10} pour les seuils critiques respectivement $\tau_{N_h} = 100$ et $\tau_{U_b} = 10$. Échelle logarithmique sur l'axe des ordonnées.

5.2, à l'aide de la stratégie d'apprentissage présentée section 5.3. La section 5.4.1 développe le protocole expérimental, en détaillant l'implémentation et les méthodes d'évaluation, et la section 5.4.2 présente les résultats expérimentaux obtenus.

5.4.1 Protocole expérimental

L'implémentation des modèles et stratégies a été réalisée avec la bibliothèque développée dans l'équipe DUKe, dédiée aux modèles graphiques probabilistes PILGRIM basée sur la bibliothèque ProBT [1]. Les paramètres utilisés pendant l'apprentissage sont un critère d'arrêt égal avec le seuil de variation de la vraisemblance 10^{-4} pour l'algorithme EM, $n_{step} = 15$ pour la stratégie *recursive* et $\mu = 0.01$ pour la stratégie *progressive*.

Les critères de suffisance δ_1 et d'indépendance δ_2 ont été estimés en appliquant les algorithmes de sur-échantillonnage du paquetage imbalanced-learn [78] suivants : ADASYN, BorderlineSMOTE, KMeansSMOTE, RandomOverSampler, SMOTE et SVMSMOTE. Les intervalles de résultats obtenus sont respectivement $[0,02; 0,12]$ pour δ_1 et $[0,34; 0,58]$ pour δ_2. Ces valeurs se situent dans la plage de valeurs où le co-training s'est avéré efficace dans notre étude précédente sur les jeux de données UCI. Cependant, les méthodes de sur-échantillonage ne sont pas utilisées pour la suite des expériences car elles ne permettent pas d'améliorer les résultats, selon une étude préliminaire.

Chaque expérience est réalisée cinq fois et applique la validation croisée à 2 échantillons

1. https ://www.probayes.com/

(*5x2-fold cross-validation*), et l'évaluation des stratégies d'apprentissage est faite par deux matrices de confusion comparant les prédictions faites par les $Model_A$ et $Model_B$ à la même vérité terrain. Les indicateurs utilisés pour évaluer les performances sont la précision équilibrée, qui est fréquemment utilisée dans des cas de jeux de données déséquilibrés, et la sensibilité. Dans notre application, la sensibilité est un indicateur très important car elle mesure à quel point le modèle a réussi à retrouver les KO, et l'impact peut être important si l'un des rares évènements très néfastes n'est pas détecté, alors qu'il provoque des dégâts importants.

Enfin, l'analyse des résultats de l'algorithme de sélection de variables utilise le package python Kneed [2], qui implémente l'algorithme Kneedle décrit dans la section 1.4.2 pour détecter le coude dans l'évolution de l'information mutuelle, et en déduire le sous-ensemble de variables à conserver.

5.4.2 Résultats

Pour l'apprentissage du modèle, quatre combinaisons d'agrégations locales et globales sont imaginables : l'agrégation local max et l'agrégation globale max (désignée comme max/max), max/max bruité, max bruité/max ou encore max bruité/max bruité. La première combinaison n'a pas d'intérêt car aucun apprentissage n'aurait lieu si toutes les prédictions sont faites de manière déterministe. La dernière est irréalisable avec l'implémentation utilisée du point de vue de la complexité. Les deux combinaisons max/max bruité, max bruité/max sont donc apprises avec les stratégies et la configuration max/max est conservée à titre comparatif.

La première section comporte une étude préliminaire visant à déterminer les a priori de Dirichlet adéquats. Après la détermination des meilleurs a priori, l'étude des performances en utilisant les seuils considérés comme optimaux permet de sélectionner la meilleure stratégie pour ce cas de figure. Enfin, l'algorithme de sélection de variables présenté dans la section 3.3.3 est appliquée sur les différents seuils disponibles, et les performances des seuils sélectionnés sont comparées avec celles des seuils optimaux.

Détermination des a priori de Dirichlet

Comme précisé dans la section 3.3.1, la méthode EAP est utilisée car elle permet d'obtenir de meilleurs résultats pour les jeux de données déséquilibrés. Il est pour cela

2. https ://github.com/arvkevi/kneed

Précision équilibrée

	u	1 ;10	1 ;20	5 ;10	5 ;20
u					
1 ;10					
1 ;20					
5 ;10					
5 ;20					

Sensibilité

	u	1 ;10	1 ;20	5 ;10	5 ;20
u					
1 ;10					
1 ;20					
5 ;10					
5 ;20					

TABLE 5.1 – Significativité des comparaisons entre les différentes paires de valeurs de α_2 pour dégradé et KO, obtenu avec un test de Student à un risque de 1%. La lettre u désigne les résultats pour les entraînements avec un *a priori* uniforme de 1 pour tous les α, c'est-à-dire $\alpha^{deg} = \{1, 1, 1\}$ et $\alpha^{KO} = \{1, 1, 1\}$. Les autres lignes correspondent à des pairs de valeurs $(\alpha_2^{deg}\,;\, \alpha_2^{KO})$. La forme complète des *a priori* est alors $\alpha^{deg} = \{1, 5, \alpha_2^{deg}\}$ et $\alpha^{KO} = \{1, 1, \alpha_2^{KO}\}$. La table de gauche indique les différences significatives pour la précision équilibrée, et la table de gauche pour la sensibilité. Vert et rouge signifient respectivement que la valeur en colonne est significativement meilleure ou pire que la valeur en ligne. Les cases grises sont les cases dont les valeurs sont non significatives.

nécessaire de déterminer les *a priori* de Dirichlet pour les distributions $P(HX_{i_d}^{\tau_{j,i}} | X_{i_d}^{\tau_{j,i}} =$ dégradé ou KO) dans le cas de l'agrégation locale en max bruité et $P(HX_{i_d} | X_{i_d} =$ dégradé ou KO) dans le cas de l'agrégation globale en max bruité. Les mêmes valeurs que dans le chapitre 3 sont envisagées.

La table 5.1 permet de savoir lesquelles de ces valeurs sont significativement meilleures que les autres en précision équilibrée et sensibilité, à l'aide d'un test de Student.

Les résultats apparaissent comme étant très différents de ceux observés dans la section 3.4.3. Il apparaît que les pairs $(\alpha_2^{deg}, \alpha_2^{KO}) = (1; 20)$ et $(5; 20)$ sont toutes deux significativement meilleures que les autres et pas meilleures l'une que l'autre en précision équilibrée. Cependant, en considérant également la sensibilité, $(5; 20)$ est significativement meilleure que toutes les autres pairs. C'est donc $\alpha_2^{deg} = 5$ et $\alpha_2^{KO} = 20$ qui sont utilisés pour le reste des expériences. En pratique, cela consiste à ajouter 5 instances artificielles où $HX_{i_d}^{\tau_{j,i}} =$ KO alors que $X_{i_d}^{\tau_{j,i}} =$ dégradé, et 20 instances artificielles où $HX_{i_d}^{\tau_{j,i}} =$ KO lorsque $X_{i_d}^{\tau_{j,i}} =$ KO au moment de l'estimation des paramètres (deuxième étape de l'algorithme *EM*).

Il est possible que cette différence de résultats marquée par rapport aux jeux de données artificiels soit due à la méthode de génération des données, qui ne représente pas fidèlement la réalité au niveau de la limite floue entre les données étiquetées dégradé ou KO.

Apprentissage commençant par la vue A (usinage)

Agr. glob.	Stratégie	**vue A** (usinage) u	s	r	p_m	p_c	**vue B** (signature broche) s	r	p_m	p_c
max	**préc. éq.**	95					77.7			
	sens.	100					83.3			
max bruité	**préc. éq.**	94.6	94.6	93.6	93.6	93.8	79	79.1	79.1	79.1
	sens.	100	100	100	100	100	83.3	83.3	83.3	83.3

Apprentissage commençant par la vue B (signature de broche)

Agr. glob.	Stratégie	**vue A** (usinage) s	r	p_m	p_c	**vue B** (signature broche) u	s	r	p_m	p_c
max	**préc. éq.**	95.2				77.7				
	sens.	100				83.3				
max bruité	**préc. éq.**	93.2	93.5	94.1	94.1	77.7	77.7	79	79	79.2
	sens.	100	100	100	100	83.3	83.3	83.3	83.3	83.3

TABLE 5.2 – Performances obtenues avec les différentes stratégies, pour chaque modèle (avec une unique agrégation globale), en commençant l'apprentissage par le modèle A (table du haut) et B (table du bas). Les entrées sont les critères avec les seuils optimaux, soit A^{120}_{rms}, V^{4}_{rms}, N^{100}_{h} et U^{10}_{b}. Les a priori utilisés pour l'apprentissage sont $\alpha_2^{deg} = 5$ et $\alpha_2^{KO} = 20$. Les cases vertes font ressortir la meilleure performance pour chaque ligne. préc. éq. signifie précision équilibrée, agr. glob. agrégation globale, u fait référence à un apprentissage non-supervisé, s à la stratégie *split*, r à la stratégie *recursive*, et p_m et p_c à la stratégie *progressive* avec les critères d'arrêts *max* et *coude*.

Performances avec les seuils optimaux

Les expériences d'apprentissage qui ne prennent en entrée que les seuils optimaux, utilisent un max ou max bruité comme agrégation globale et il n'y a aucune variable à agréger localement (voir figure 5.4).

La table 5.2 regroupe les résultats des stratégies lorsque l'apprentissage commence respectivement par le modèle A et le modèle B, avec en entrée les seuils considérés comme optimaux. Les cases vertes font ressortir la meilleure performance par ligne. Pour le modèle A, les résultats sont favorables à l'agrégation max, l'utilisation du max bruité diminue les performances. En revanche, pour le modèle B, les meilleures performances sont obtenues par la stratégie *progressive*, avec le critère d'arrêt p_c. Les résultats en sensibilité sont toujours les mêmes quelle que soit la stratégie ou l'agrégation.

Ces résultats sont encourageants et montrent que la structure proposée permet de retrouver efficacement les incidents, malgré le taux de déséquilibre très important.

Nouvelle identification des seuils critiques

Une première sélection de variables est lancée avec la stratégie progressive (critère d'arrêt *maximum*), une agrégation locale en max bruité et une agrégation globale en max, le coût de calcul ne permettant pas d'utiliser un max bruité pour les deux agrégations. Cette première série d'expériences avec 7 seuils pour chaque critère permet de déterminer l'itération de *progressive* qui subit le coude sur l'évolution de la confiance moyenne dans ce contexte expérimental. La figure 5.10 montre l'évolution de la confiance moyenne en les étiquettes ajoutées à chaque itération et le point rouge a pour abscisse l'itération 86, à laquelle se produit la plus grande chute de confiance. L'apprentissage est par la suite relancé avec la stratégie *progressive* pour 86 itérations.

L'importance des seuils est déterminée à l'aide de l'algorithme de sélection de variables par élimination de la section 3.3.3. La figure 5.11 montrent les diagrammes en violon de la distribution du rang de suppression des variables pendant l'algorithme de sélection de variables par élimination pour chaque critère avancé. Les variables sont classées dans l'ordre décroissant du rang moyen de suppression, par conséquent, les variables de gauche correspondent aux variables supprimées en moyenne en dernières, et celles de droite en premières. L'ordre de suppression des variables donne une indication sur l'importance de la variable dans le diagnostic : plus une variable est supprimée tardivement (rang élevé), plus son importance est grande. Ces résultats montrent que les critères A_{rms} et V_{rms} sont majoritairement considérées comme plus utiles à la prédiction que N_h et U_b, à

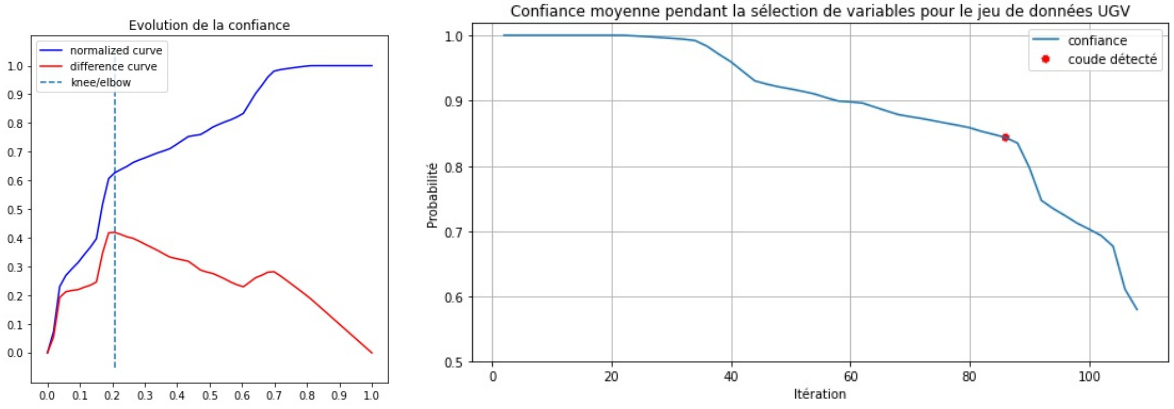


FIGURE 5.10 – A gauche, la sortie de la fonction plot_knee_normalized() pour la détection du coude sur l'évolution de la confiance moyenne pendant la sélection de variables. A droite, la courbe d'évolution de la confiance moyenne à chaque itération pendant la sélection de variables. Le point rouge correspond à l'itération de détection du coude.

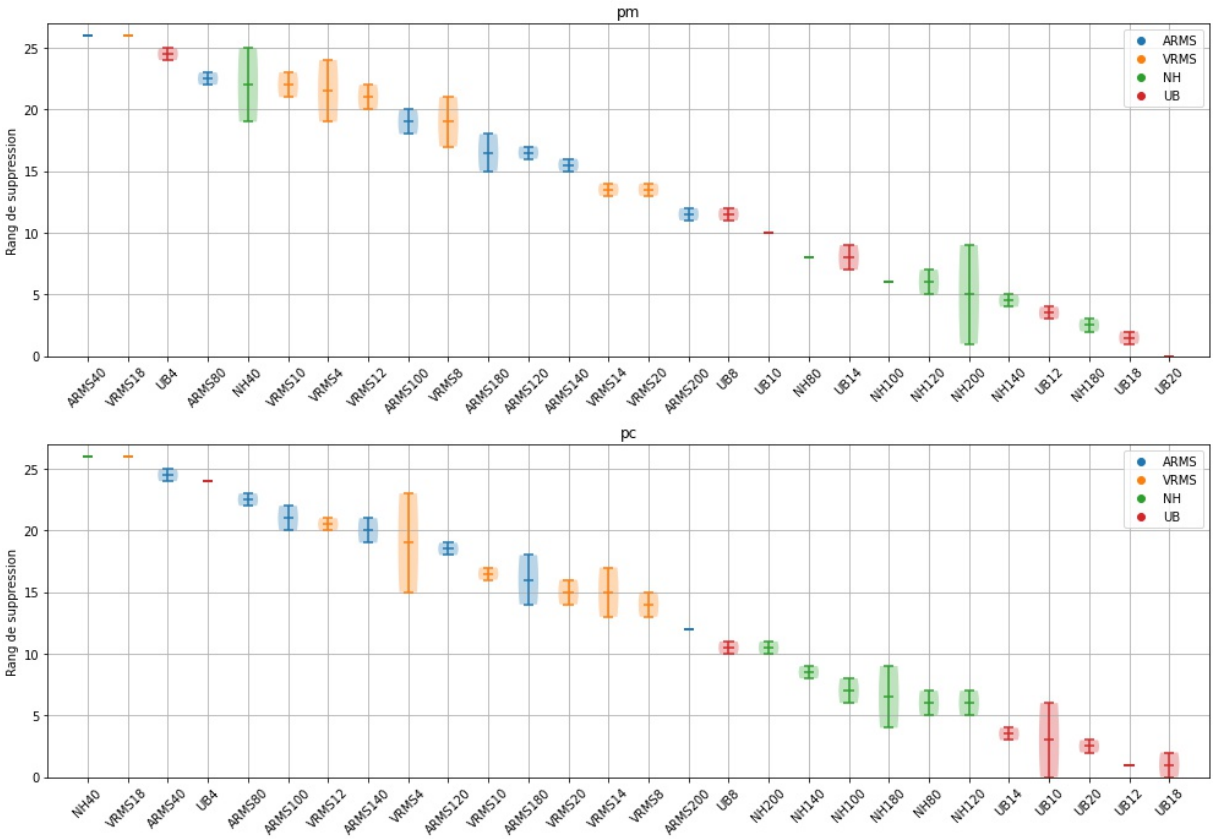


FIGURE 5.11 – Diagrammes en violon représentant la distribution du rang de suppression des variables pendant l'algorithme de sélection de variables par élimination pour chaque critère avancé. La sélection est faite avec la stratégie *progressive* et le critère d'arrêt *max* (graphe du haut) et *coude* (graphe du bas), et les diagrammes sont classés dans l'ordre décroissant du rang moyen de suppression de la variable lors de la procédure de sélection de variables.

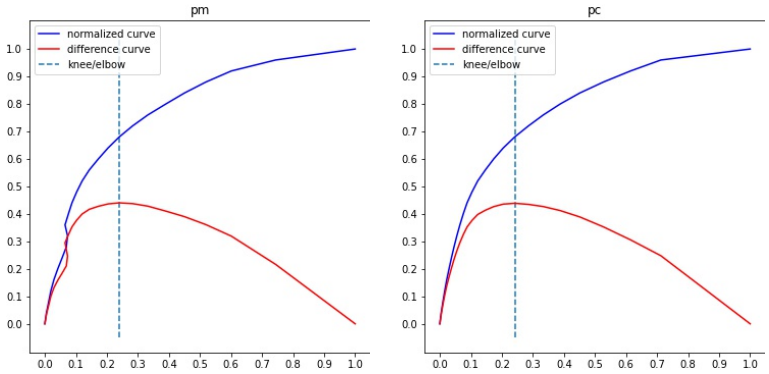


FIGURE 5.12 – Sortie de la fonction plot_knee_normalized() du package kneed pour les deux détections de coude par l'algorithme Kneedle, pour les sélections de variables avec p_m et p_c.

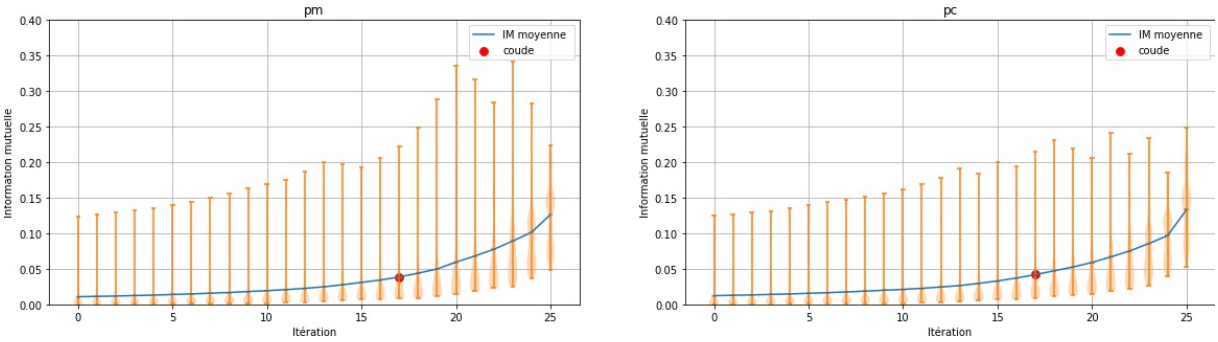


FIGURE 5.13 – Diagrammes en violon (en jaune) représentant la distribution de l'information mutuelle à chaque itération de l'algorithme de sélection de variables. La sélection est faite avec la stratégie *progressive* et les critères d'arrêts *max* et *coude* (diagrammes de gauche et droite respectivement). Le point rouge désigne le coude détecté par l'algorithme Kneedle

l'exception de N_h^{40} et U_b^4, qui sont tous les deux dans les variables considérées comme les plus importantes quel que soit le critère d'arrêt (p_m ou p_c).

La figure 5.12 montre la courbe normalisée générée par l'algorithme kneedle, et la courbe déduite de la différence entre la vraie courbe et la ligne tracée entre les points extrêmes, tel que dans la figure 1.11. Le coude détecté pour chaque stratégie p_m et p_c est ensuite rapporté sur les courbes de la figure 5.13 (points rouges). Le point détecté est celui d'abscisse 17 pour chacune des sélections de variables, c'est-à-dire que la suppression de variables doit s'arrêter à l'itération 17. Par conséquent, le sous-ensemble de variables déterminé comme étant celui à conserver pour p_m est le suivant (par ordre d'importance) : A_{rms}^{40}, V_{rms}^{18}, U_b^4, A_{rms}^{80}, N_h^{40}, V_{rms}^{10}, V_{rms}^4, V_{rms}^{12}, A_{rms}^{100}, V_{rms}^8, et pour p_c : N_h^{40}, V_{rms}^{18},A_{rms}^{40}, U_b^4,

	p_m	p_c
A_{rms}^{120}	11	10
V_{rms}^{4}	7	9
N_{h}^{100}	21	20
U_{b}^{10}	18	25

TABLE 5.3 – Rang d'importance (inverse du rang de suppression des figures 5.11) des critères déterminés comme optimaux par Godreau et al [41]. Résultats issus de la sélection de variables faite avec la stratégie *progressive* et les critères d'arrêts *max* et *coude* : p_m et p_c respectivement.

A_{rms}^{80}, A_{rms}^{100}, V_{rms}^{12}, A_{rms}^{140}, V_{rms}^{4} et A_{rms}^{120}.

Les critères trouvés par Godreau et al [41], considérés précédemment comme étant optimaux sont A_{rms}^{120}, V_{rms}^{4}, N_{h}^{100} et U_{b}^{10}, l'optimisation utilisée n'autorisant le choix que d'un seul seuil par critère. Le rang d'importance, déterminé par le rang dans le classement inverse de l'ordre moyen de suppression des variables est donné par la table 5.3 pour les critères considérés précédemment comme optimaux. Le critère V_{rms}^{4} est considéré comme important par les deux exécutions de la sélection de variables par suppression, et c'est le seul qui soit inclus dans le sous-ensemble de variables sélectionné, avec A_{rms}^{120} pour la sélection avec p_c. A_{rms}^{120} fournit une information assez proche de A_{rms}^{80}, A_{rms}^{100} et A_{rms}^{140}, le fait que ces variables soient assez élevées dans le classement de p_c et p_m indique une cohérence avec la sélection de Godreau et al [41]. En revanche, N_{h}^{100} et U_{b}^{10} sont considérés comme étant les moins importants par l'algorithme de sélection de variables par suppression.

La consultation d'un avis d'expertise sur le sous-ensemble de seuils sélectionnés révèle que celui-ci est cohérent mais qu'il présente cependant des redondances en terme d'information, notamment entre A_{rms}^{80}, A_{rms}^{100} et A_{rms}^{120}, et entre V_{rms}^{12} et V_{rms}^{18}. Il est possible que ces variables avec des informations similaires ne soient pas sélectionnés simultanément si la procédure de sélection de variables prenait en compte l'information mutuelle entre les entrées, en plus de l'information mutuelle de chaque entrée et de la sortie. Cela permettrait de sélectionner un sous-ensemble de variables plus restreint et avec moins de redondances.

Performances avec les nouveaux seuils critiques

Les résultats obtenus à partir des nouveaux seuils critiques sélectionnés sont comparés dans cette section aux résultats des seuils optimaux précédents. La table 5.4 regroupe les résultats des stratégies lorsque l'apprentissage commence respectivement par le modèle A et le modèle B, avec en entrée les seuils sélectionnés (voir figure 5.14). Les cases vertes

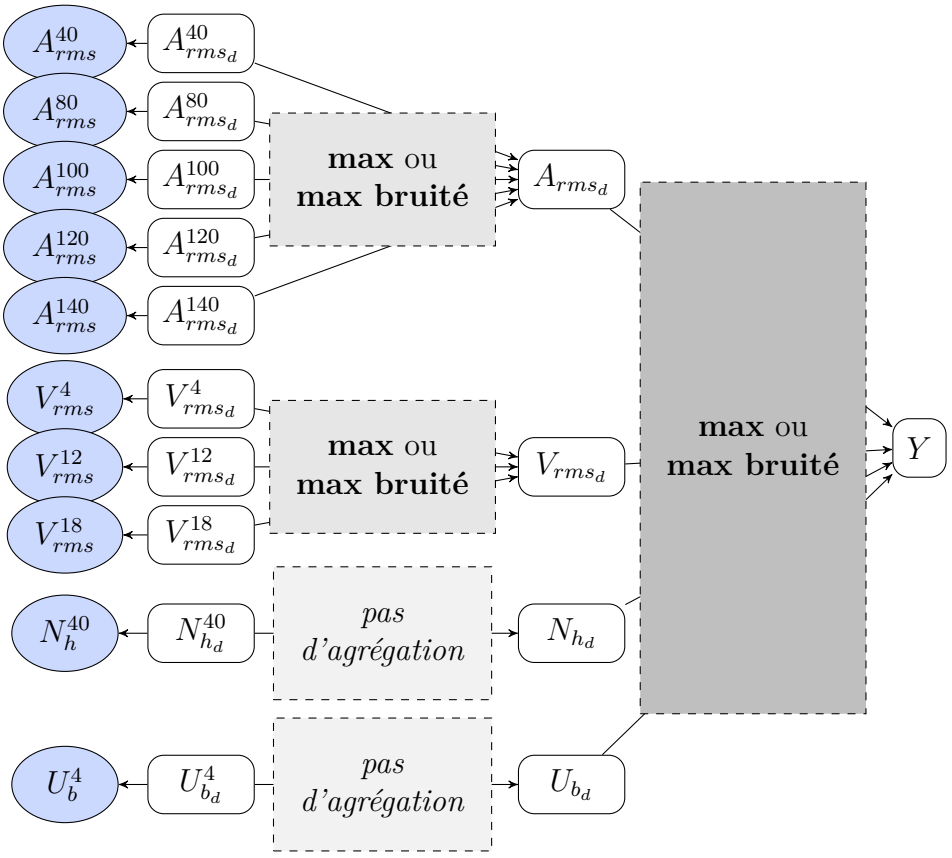


FIGURE 5.14 – Architecture du modèle de la surveillance du procédé d'usinage après la sélection des seuils avec p_c.

Début vue A

Agr. loc/glob	Stratégie	vue A u	vue A s	vue A r	vue A p_m	vue A p_c	vue B s	vue B r	vue B p_m	vue B p_c
max/max	**préc. éq.**	81.9					77.6			
	sens.	100					83.3			
max br./max	**préc. éq.**	81.9	81.9	71.4	71.9	72.2	76.9	76.5	76.5	76.5
	sens.	100	100	100	100	100	83.3	91.7	91.7	91.7
max/max br.	**préc. éq.**	81.9	81.9	63.4	58.1	57.1	79.8	66	60.6	60.3
	sens.	100	100	100	100	100	66.7	94.2	100	100

Début vue B

Agr. loc/glob	Vue Stratégie	A s	A r	A p_m	A p_c	B u	B s	B r	B p_m	B p_c
max/max	**préc. éq.**	81.9				77.6				
	sens.	100				83.3				
max br./max	**préc. éq.**	72.6	71.5	71.7	72.6	77.6	77.6	76.5	78.1	77.1
	sens.	100	100	100	100	83.3	83.3	91.7	95.8	86.1
max/max br.	**préc. éq.**	82.7	82.7	57.9	58.7	77.6	77.6	77.6	59.7	60.2
	sens.	100	100	100	100	83.3	83.3	83.3	100	100

TABLE 5.4 – Performances obtenues avec les critères des seuils sélectionnés en commençant l'apprentissage par le modèle A (table du haut) et B (table du bas), effectué avec deux agrégations, une locale et une globale, (Agr. loc/glob) et les a priori $\alpha_2^{deg} = 5$ et $\alpha_2^{KO} = 20$. Les cases vertes font ressortir la meilleure performance pour chaque ligne.

font ressortir la meilleure performance par ligne.

Les performances atteintes en agrégation max bruité/max de la stratégie simple non-supervisée sont généralement les mêmes que celles de l'agrégation max/max (qui est totalement déterministe et n'est pas impactée par les stratégies de co-training). Celles-ci sont les meilleures performances atteintes pour le modèle A, et pour l'optimisation de la précision équilibrée du modèle B, mais les meilleures performances en sensibilité pour le modèle B sont obtenues par p_m. Par conséquent, en agrégation max bruité/max, les stratégies de co-training dégradent les performances par rapport à un apprentissage non-supervisé simple, sauf pour la sensibilité du modèle B.

En agrégation max/max bruité, la précision équilibrée n'est pas non plus améliorée par l'utilisation du co-training. En revanche, la stratégie *progressive* permet d'atteindre une sensibilité de 100% en moyenne, ce qui signifie que tous les incidents sont toujours retrouvés. Cette observation est concordante avec les conclusions du chapitre 4 : *progressive* permet généralement d'améliorer la sensibilité par rapport aux autres stratégies. La comparaison entre la table 5.2 et la table 5.4 permet de conclure que sur ce jeu de données,

la sélection des seuils a permis également un gain important dans les différents indicateurs de performances, et notamment la sensibilité, qui atteint en moyenne les 100% pour le modèle B.

Il est important de rappeler que les stratégies sont évaluées avec une vérité dont les KO identifiés ont été détectés à partir des valeurs des critères considérés comme optimaux : A_{rms}^{120}, V_{rms}^{4}, N_{h}^{100} et U_{b}^{10}. Comme mentionné précédemment, il est donc possible que certains défauts n'aient pas encore été identifiés, et en particulier, il est possible que ceux-ci soient identifiables uniquement avec les nouveaux seuils sélectionnés. Il est donc envisageable que la baisse de précision équilibrée constatée précédemment soit liée à une détection de nouveaux défauts par le modèle, qui seraient évalués ici comme une erreur. La suite de l'étude propose de rechercher ces éventuels endommagements non identifiés, en utilisant les nouvelles variables sélectionnées.

5.5 Nouveaux endommagements

Dans cette partie, la stratégie progressive est utilisée, avec le critère d'arrêt p_c, l'agrégation locale max bruité et l'agrégation globale max. Les résultats des nouveaux modèles et l'étude des endommagements prédits par ceux-ci à partir des seuils sélectionnés ont permis d'identifier trois nouveaux endommagements non détectés précédemment, aux jours 47, 114 et 151, qui ont tous été vérifiés par expertise. Comme mentionné précédemment, certains des critères sélectionnés présentent de grandes similarités les uns avec les autres, et ils permettent de trouver les mêmes défauts. Dans les figures suivantes, V_{rms}^{12}, A_{rms}^{80}, A_{rms}^{120}, A_{rms}^{140} ne sont pas affichées car elles n'apportent pas d'informations supplémentaires par rapport aux autres variables, pour les défauts présentés. Les figures 5.15 montre les données collectées pendant les vies des broches 1 et 2 pour les vues A et B. Les traits verticaux bleus sont indiqués sur les jours détectés comme KO par le modèle et étant des endommagements déjà connus (trouvés par Godreau et al [41]). Les croix bleues et les triangles bleus représentent respectivement les KO et les dégradé détectés par chaque critère pour ces jours donnés. Les traits verticaux verts sont indiqués sur les jours détectés comme KO par le modèle et étant des endommagements nouvellement détectés. Les croix rouges et les triangles oranges représentent respectivement les KO et les dégradé détectés par chaque critère pour ces jours donnés.

Deux des nouveaux endommagements identifiés se sont produits pendant la vie de la broche 1 ; le jour 47, que de nombreux critères ont indiqué comme KO. Les valeurs élevées

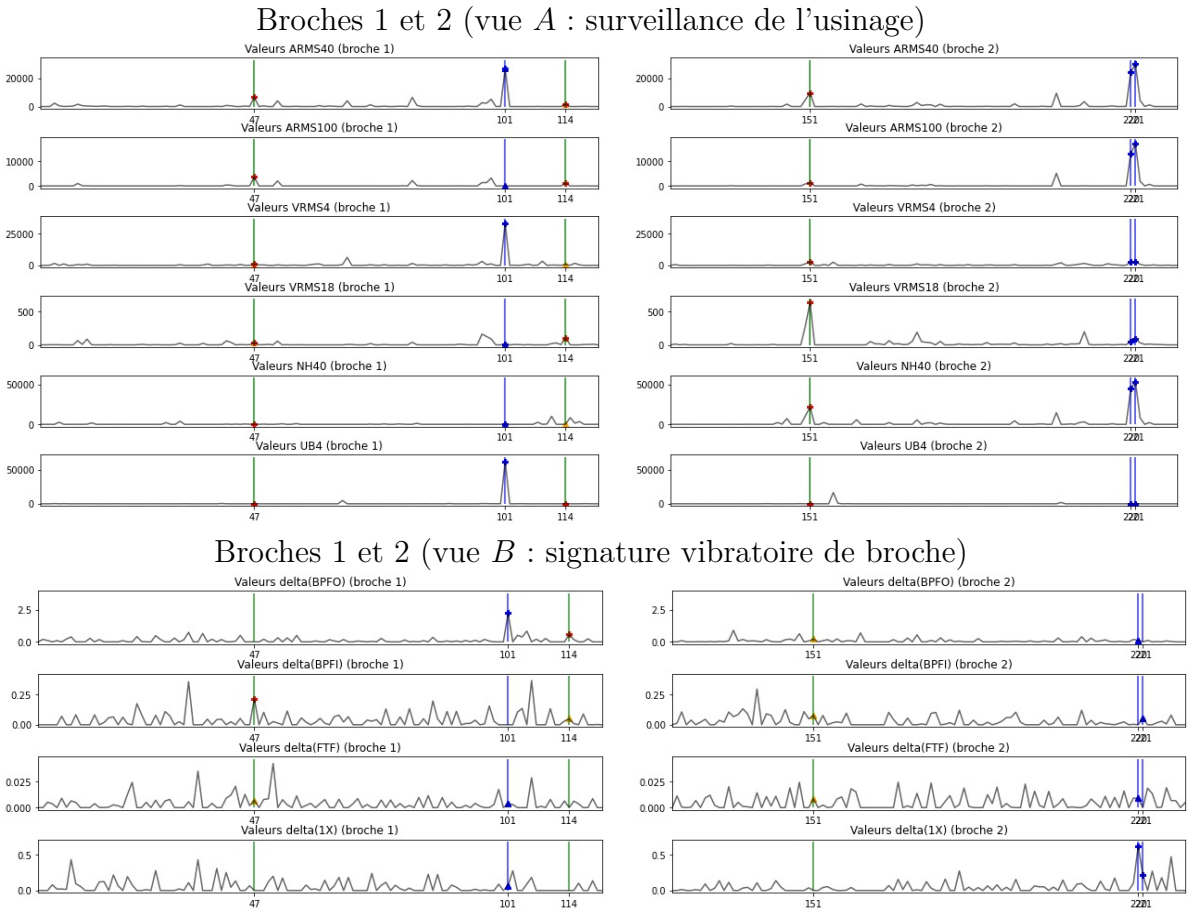
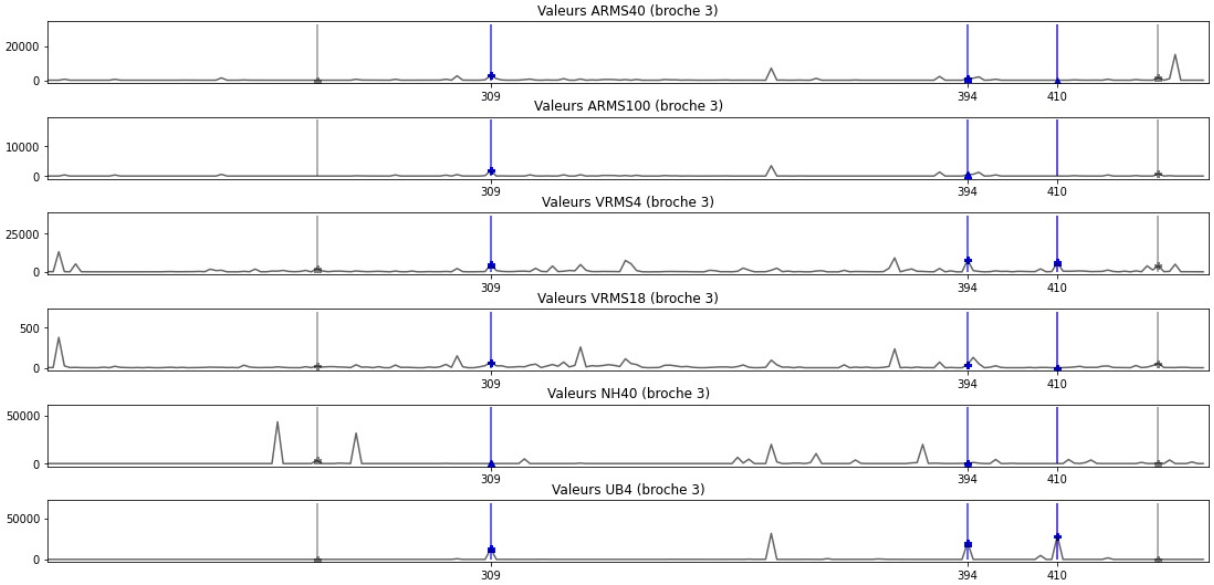


FIGURE 5.15 – Données récoltées pendant la vie de la première et de la deuxième broche. En haut : pendant le procédé d'usinage, vue A (seuils sélectionnés), en bas : pendant la signature vibratoire de la broche.
Les traits verticaux bleus sont indiqués sur les jours détectés comme KO par le modèle et étant des endommagements déjà connus (trouvés par Godreau et al [41]). Les croix bleues et les triangles bleus représentent respectivement les KO et les dégradé détectés par chaque critère pour ces jours donnés.
Les traits verticaux verts sont indiqués sur les jours détectés comme KO par le modèle et étant de nouveaux endommagements détectés. Les croix rouges et les triangles oranges représentent respectivement les KO et les dégradé détectés par chaque critère pour ces jours donnés.

Broche 3 (vue A : surveillance de l'usinage)



Broche 3 (vue B : signature vibratoire de broche)

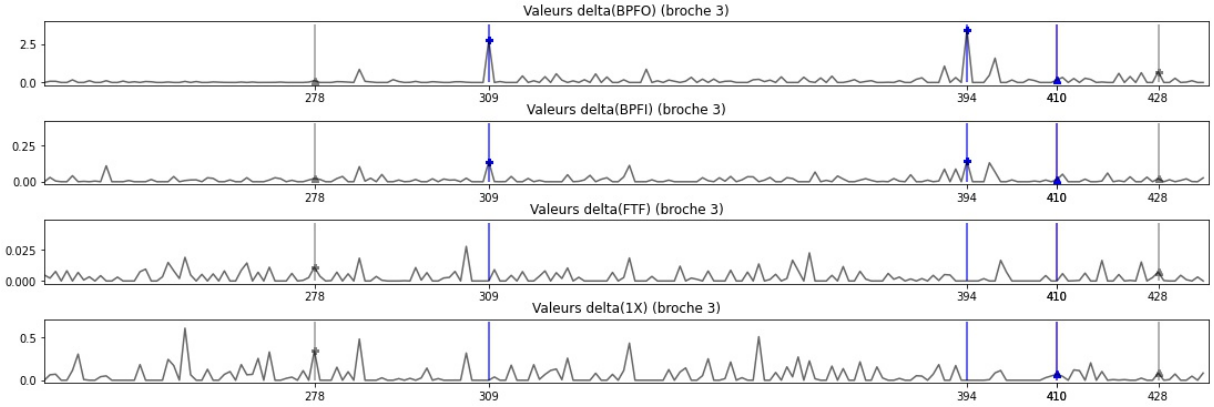


FIGURE 5.16 – Données récoltées pendant la vie de la troisième broche. En haut : pendant le procédé d'usinage, vue A (seuils sélectionnés), en bas : pendant la signature vibratoire de la broche.
Les traits verticaux bleus sont indiqués sur les jours détectés comme KO par le modèle et étant des endommagements **déjà** connus (trouvés par Godreau et al [41]). Les croix bleues et les triangles bleus représentent respectivement les KO et les dégradé détectés par chaque critère pour ces jours donnés.
Sur la vue B, le trait vertical rouge correspond au jour 410, pendant lequel un énorme bris s'est produit, mais aucun endommagement de la broche n'est détecté sur ces données car celle-ci avait déjà été abîmée auparavant. Les traits gris sont des KO détectés par le modèle B aux jours 278 et 428 mais non validés par le modèle A.

de A^{40}_{rms} et A^{100}_{rms} permettent de confirmer l'occurrence d'un événement anormal pouvant avoir endommagé la broche. Cette observation est confirmée par une valeur très élevée de $\Delta(BPFI)$, ce qui révèlent un endommagement des roulements intérieurs. Le jour 114 a été étiqueté comme KO par plusieurs variables et notamment V^{18}_{rms}. La détection d'un pic par cette variable indique un niveau vibratoire trop élevé, et un endommagement est révélé sur les roulements extérieurs grâce à $\Delta(BPFO)$.

Enfin, un dernier endommagement est confirmé pendant la vie de la broche 2, au jour 151, avec une vibration extrêmement violente détectée par V^{18}_{rms}, et un pic modéré pour $\Delta(BPFI)$ et $\Delta(FTF)$. Tous les défauts précédemment annotés par Godreau et al [41] pendant les vies des broches 1 et 2 sont retrouvés : au jour 101 grâce à A^{40}_{rms}, V^{4}_{rms} et U^{18}_{b} du modèle A, et $\Delta(BPFO)$ du modèle B, et le jour 220 grâce à A^{40}_{rms} et A^{100}_{rms}, et N^{40}_{h} du modèle A, et $1X$ du modèle B.

Les données récoltées pendant la vie de la troisième broche sont représentées sur la figure 5.16. Aucun nouveau défaut n'a été détecté pendant celle-ci, cependant il est indiqué que U^{4}_{b} détecte efficacement les trois bris précédemment étiquetés par Godreau et al [41] aux jours 309, 394 et 410. Le jour 410 n'est pas détecté par le modèle B, ce qui est cohérent car le bris de ce jour n'a pas endommagé la broche, qui était déjà très endommagée par les événements des jours 309 et 394.

Enfin, les traits gris de la figure 5.16 sont des KO détectés par le modèle B aux jours 278 et 428 mais non détectés par le modèle A. Ces endommagements ne sont pas validés par expertise pour cette raison. Cependant, il serait intéressant de les explorer en détails car ils seraient peut-être détectables par un autre critère de la vue A, qui n'est pas inclus dans les critères sélectionnés, ou bien même directement dans les données brutes dans le cas où l'événement est si court qu'il n'est pas visible avec les critères avancés.

5.6 Conclusion

Ce chapitre rassemble les contributions des deux chapitres précédents : le modèle dédié au diagnostic présenté dans le chapitre 3, et les stratégies de co-training non-supervisé décrites dans le chapitre 4. L'ensemble permet de valider leur fonctionnement sur une application réelle du domaine industriel, avec le diagnostic de l'électrobroche en usinage à grande vitesse.

L'étude repose sur un ensemble de données récolté en conditions réelles de production, qui reprend l'ensemble des éléments décrits dans la partie 5.1, avec au total trois durées de

vies de broches. Les expériences ont permis de valider l'utilité des stratégies de co-training sur des données réelles dans certains cas. L'apprentissage avec la stratégie *progressive* permet l'amélioration de la sensibilité du modèle B, qui est l'indicateur le plus important dans ce cas d'étude.

La sélection de variables a permis de déterminer, parmi vingt-huit critères avancés, dix critères parmi les plus utiles à la prédiction. Les performances obtenues à la suite de cette sélection sont meilleures, en comparaison, que celles obtenues avec le même modèle et les mêmes stratégies mais des seuils sélectionnés précédemment par une autre méthode. L'intérêt de l'architecture proposée est donc de se passer de toute connaissance préalable.

Enfin, l'application des modèles de diagnostic et des stratégies de co-training avec les nouveaux seuils sélectionnés a permis de retrouver l'ensemble des événements et dégradations annotées par Godreau et al [41], ainsi que d'en identifier trois nouvelles.

Le travail présenté dans ce chapitre peut être amélioré de plusieurs façons. L'analyse par expertise de la sélection des seuils indique que les seuils sélectionnés sont pertinents, et cette conclusion est confirmée par le fait que de nouveaux défauts sont détectés grâce aux nouveaux seuils sélectionnés. Cependant, il ressort qu'une sélection de 10 critères est excessive car certains d'entre eux sont très semblables et permettent de détecter les mêmes endommagements de l'électrobroche. Ce défaut peut être corrigé en ajoutant à l'algorithme de sélection de variables une prise en compte de l'information mutuelle entre les différentes variables d'entrées, en plus de l'information mutuelle entre chaque entrée et la sortie.

De plus, une partie des données disponibles n'est pas exploitée pour le modèle B ; les signatures vibratoires de broche issues des différentes vitesses de rotation de l'électrobroche. Ces données pourraient poursuivre l'étude débutée dans ce chapitre, en effectuant une série d'expériences avec les différentes agrégations locales et globales, en prenant en entrée plusieurs vitesses de rotation plutôt que seulement celle à la vitesse de 17 500 tours par minute. L'utilisation de l'algorithme proposé de sélection de variables sur ces données pourrait faire ressortir de nouvelles vitesses de broches complémentaires à inclure dans le modèle pour la prédiction de l'état de la broche.

Enfin, les KO détectés par le modèle B aux jours 278 et 428 ne sont pas détectés par le modèle A. Il serait possible que l'exploration plus approfondie des données de la vue A dans ce sens permette leur validation comme événement ayant dégradé la broche. Ils pourraient être détectables par un autre critère de la vue A, ou bien directement dans les données brutes dans le cas où l'événement est si court qu'il n'est pas visible avec les

critères avancés.

En résumé

- ✓ L'algorithme de sélection automatique des variables permet la détection de sous-ensemble qui permet la détection d'autres endommagements et de se passer d'une pré-sélection des seuils à l'aide d'une autre méthode ;
- ✓ Le co-training non-supervisé permet une amélioration de la performance en sensibilité pour le modèle B sur le jeu de données d'UGV en comparaison avec un apprentissage simple non-supervisé ;
- ✓ Le modèle présenté et les stratégies utilisées permettent une conservation de la sensibilité ; tous les défauts précédemment identifiés sont retrouvés ;
- ✓ Trois nouveaux endommagements ont été détectés par les deux modèles et validés par expertise.

CONCLUSION

Conclusion

Un ensemble de méthodes pour la détection de problèmes et le diagnostic sont proposées dans cette thèse. L'apprentissage non-supervisé des paramètres de réseaux bayésiens, le traitement des données de la sortie des capteurs jusqu'à la prédiction finale, et la sélection des critères les plus importants pour celle-ci font partie des thèmes étudiés dans ce manuscrit.

Le chapitre 3 propose une architecture de réseau bayésien adapté au diagnostic à partir de capteurs ainsi qu'une méthode d'apprentissage de ce modèle, et avance également une procédure de sélection des critères importants pour un problème de diagnostic donné. Ce modèle est générique et utilisable dans un contexte supervisé ou non-supervisé. La méthode de sélection de variables est généralisable à d'autres modèles et ne se limite pas à un contexte dans lequel une valeur élevée correspond à un problème. Cet ensemble de contributions est validé par un apprentissage supervisé dans la section expérimentale, qui prouve la capacité de l'algorithme de sélection à identifier les variables les plus utiles pour la prédiction, ainsi que ses performances en tant que classifieur dans un contexte supervisé, et ce même sur des jeux de données très déséquilibrés.

Les travaux du chapitre 4 ont permis d'explorer le domaine prometteur du co-training, et de proposer trois stratégies génériques non supervisées, ce qui est très rare dans ce domaine. Ces stratégies sont théoriquement utilisables sur tout classifieur probabiliste, mais n'ont été testées que sur le classifieur bayésien naïf. Des données artificielles et des données UCI ont été utilisées pour les expérimentations. Chacune des stratégies a ses avantages : la stratégie *split* est particulièrement rapide, *recursive* permet dans un certain nombre de cas une grande augmentation de la précision équilibrée, *progressive* au

contraire donne plutôt une amélioration de la sensibilité. Les inconvénients des stratégies ont également été bien identifiés, *split* offre des résultats peu précis et peu significativement différents par rapport un apprentissage simple non supervisé, *recursive* a tendance à entrer pour certains benchmarks dans une boucle de rétroaction négative (généralement corrigée par *progressive*), et *progressive* connaît fréquemment une baisse de performances à la fin de son apprentissage, qui peut être évité grace au critère d'arrêt p_c. De manière générale, il a été observé que p_m donne le plus d'améliorations significatives des performances sur les jeux de données étudiés. L'étude des hypothèses de suffisance et d'indépendance est une part importante du co-training et de nombreuses études se proposent de trouver un cadre de référence pour son utilisation. Ling et al [81] énonce dans ses conclusions que le co-training fonctionne sur des jeux de données quand $\delta_1 < 0.23$ et $\delta_2 < 0.15$. Les jeux de données utilisés pour l'étude présentée dans ce manuscrit sortent de ce cadre et les stratégies parviennent malgré tout à tirer partie des informations pour améliorer les prédictions.

Le chapitre 5 rassemble ces contributions : le modèle dédié au diagnostic et la sélection de variables présentés dans le chapitre 3, et les stratégies de co-training non-supervisé décrites dans le chapitre 4. L'ensemble permet de valider leur fonctionnement sur une application réelle du domaine industriel, avec le diagnostic de l'électrobroche en usinage à grande vitesse. L'étude repose sur un ensemble de données récolté en conditions réelles de production, qui reprend l'ensemble des éléments décrits dans la partie 5.1, avec au total trois durées de vies de broches. Les stratégies de co-training ont pu améliorer les détections de problèmes, notamment pour le modèle entraîné avec les données issues des signatures vibratoires de broche. De plus, un sous-ensemble de variables parmi les critères avancés a été proposé par l'algorithme de sélection de variables et validé comme étant pertinent par une analyse d'expertise, le tout de façon non-supervisée. L'architecture proposée peut donc se passer de toute connaissance préalable sur les meilleurs critères à utiliser. L'apprentissage des modèles de diagnostic et des stratégies de co-training avec les nouveaux seuils sélectionnés a permis de retrouver l'ensemble des événements et dégradations annotées par Godreau et al [41], ainsi que d'en identifier trois nouvelles.

Perspectives à court terme

La capacité à sélectionner les variables les plus pertinentes de l'architecture proposée est validée sur des jeux de données simples. Il serait intéressant d'évaluer la capacité du

modèle à retrouver les variables pertinentes avec des jeux de données plus complexes, contenant plus de critères et plus de seuils. A la suite de la sélection de variables, il serait également possible d'effectuer une deuxième série d'expériences pour évaluer la capacité du modèle à la détection de problème avec le sous-ensemble sélectionné avec une agrégation en max bruité, qui était impossible avec toutes les variables en raison des limites de notre implémentation. Les expériences sur l'algorithme de sélection de variables s'étant faite dans un contexte supervisé, valider la même série d'expériences avec un apprentissage non-supervisé permettrait d'étendre les possibilités d'utilisation de cette architecture.

Les stratégies de co-training non-supervisé proposées sont génériques et utilisables par tout classifieur probabiliste mais n'ont été testées que sur le classifieur bayésien naïf. Il serait enrichissant pour la suite de reproduire les expériences avec d'autres classifieurs reconnus tel que le TAN, le BAN ou un autre modèle cité dans la section 1.4.1, pour voir si les mêmes conclusions ressortent.

Il est possible de compléter la série d'expérience sur les stratégies de co-training en modifiant les critères d'arrêt des stratégies *progressive* ou *recursive*. Pour certains jeux de données, p_c améliore les résultats mais dégrade les performances pour d'autres. Il a été identifié que ces baisses de précision et sensibilité sont potentiellement dues à un arrêt de l'apprentissage parfois trop tardif. Il est possible qu'un critère d'arrêt autre que la recherche d'un coude dans la confiance d'ajout en les étiquettes soit plus pertinent, par exemple un seuil de confiance sous lequel ne pas tomber. La stratégie *recursive* pourrait également bénéficier d'un autre critère d'arrêt qu'un nombre d'itérations fixé à l'avance, par exemple surveiller la vraisemblance et arrêter les itérations lorsque celle-ci a cessé de s'améliorer de manière significative.

Le travail sur le cas d'étude de l'UGV présenté dans le chapitre 5 peut-être enrichi en exploitant les signatures vibratoires de broche issues des différentes vitesses de rotation de l'électrobroche. Ces données pourraient poursuivre l'étude de ce chapitre, en effectuant une série d'expériences avec les différentes agrégations locales et globales, en prenant en entrée plusieurs vitesses de rotation plutôt que seulement celle à la vitesse de 17 500 tours par minute. L'utilisation de l'algorithme proposé de sélection de variables sur ces données pourrait faire ressortir de nouvelles vitesses de broches complémentaires à inclure dans le modèle pour améliorer la prédiction de l'état de la broche. Enfin, les KO détectés par le modèle B (signature vibratoire de la broche) aux jours 278 et 428 ne sont pas détectés par le modèle A. Il est possible que l'exploration plus approfondie des autres critère de la vue A (surveillance de l'usinage), ou bien directement dans les données brutes, permette

leur validation comme événement ayant dégradé la broche.

Perspectives à moyen et long terme

La méthode de sélection de variables permet de retrouver les seuils utiles à la prédiction, mais certaines imprécisions ont été relevées dans la section 3.4.3 où certains seuils sont sélectionnés à la place d'autres, et la section 5.4.2 soulève le problème de redondance d'information entre certains des critères sélectionnés. Une façon d'améliorer la méthode de sélection de variables est de prendre en compte l'information mutuelle entre les variables d'entrée, en plus de l'information mutuelle entre chaque entrée et la sortie, tel que proposé par Battiti [8] ou Hoque et al [54]. L'architecture proposée pour le diagnostic pourrait également être étendue à un classifieur à plusieurs classes plutôt que de binariser la prédiction.

Certaines imprécisions des algorithmes de co-training pourraient être améliorés. La boucle de rétroaction négative observée lors de certaines expériences est partiellement corrigée par la stratégie *progressive*, mais il serait possible de raffiner plus encore le passage d'information en ne transférant pas l'étiquette de la classe la plus certaine, mais la distribution de probabilité de l'état comme une *soft evidence* [91], comme proposé dans le contexte multi-agent par Vomlel [129]. De plus, l'étude sur le respect des hypothèses intrinsèques au co-training montre que l'hypothèse de suffisance n'est pas toujours respectée dans les applications réelles, il serait possible d'étendre ce travail sur le co-training non supervisé avec des vues insuffisantes, comme proposé par Guo et al [46], ou encore d'améliorer la méthode de séparation des vues pour améliorer le respect des hypothèses intrinsèques au co-training.

Enfin, des poursuites plus large de ces travaux peuvent être imaginées. La modélisation actuelle se détache de la dimension temporelle après le seuillage des critères, au moment de l'agrégation des critères bruts en critères avancés. En effet, cette agrégation a pour objectif de détecter des événements à haut niveau dans le signal brut. Ces événements sont porteurs de sens au niveau du processus, ils peuvent révéler des vibrations intenses, un broutement, un bris d'outil. Il pourrait être intéressant de revenir aux données brutes après avoir identifié les événements à un haut niveau avec les réseaux bayésiens, pour ensuite étudier la dynamique temporelle de ces événements et comprendre comment l'apparition d'un événement favorise le déclenchement d'un autre. Cela pourrait être réalisé par exemple avec des modèles graphiques d'événements (*GEM, graphical event models*)

[45, 86], qui peuvent être appris avec des fichiers de journaux d'événements, et détecter des enchaînements d'événements à plus ou moins long terme.

TABLE DES FIGURES

LISTE DES TABLEAUX

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] Steven ABNEY, « Bootstrapping », in : *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, p. 360-367.

[2] P.A. AGUILERA, A. FERNÁNDEZ, R. FERNÁNDEZ, R. RUMÍ et A. SALMERÓN, « Bayesian networks in environmental modelling », in : *Environmental Modelling & Software* 26.*12* (2011), p. 1376-1388, ISSN : 1364-8152, DOI : https://doi.org/10.1016/j.envsoft.2011.06.004, URL : https://www.sciencedirect.com/science/article/pii/S1364815211001472.

[3] Pedro Aguilera AGUILERA, Antonio FERNÁNDEZ, Rosa FERNÁNDEZ, Rafael RUMIÉ et Antonio SALMERÓN, « Bayesian networks in environmental modelling », in : *Environmental Modelling & Software* 26.*12* (2011), p. 1376-1388.

[4] Paul ARORA, Devon BOYNE, Justin J. SLATER, Alind GUPTA, Darren R. BRENNER et Marek J. DRUZDZEL, « Bayesian Networks for Risk Prediction Using Real-World Data : A Tool for Precision Medicine », in : *Value in Health* 22.*4* (2019), p. 439-445, ISSN : 1098-3015, DOI : https://doi.org/10.1016/j.jval.2019.01.006, URL : https://www.sciencedirect.com/science/article/pii/S1098301519300579.

[5] Mohamed Amine ATOUI, Achraf COHEN, Sylvain VERRON et Abdessamad KOBI, « A single Bayesian network classifier for monitoring with unknown classes », in : *Eng. Appl. of AI* 85 (2019), p. 681-690, DOI : 10.1016/j.engappai.2019.07.016, URL : https://hal.univ-angers.fr/hal-02489918.

[6] Maria-Florina BALCAN, Avrim BLUM et Ke YANG, « Co-training and expansion : Towards bridging theory and practice », in : *Advances in neural information processing systems*, 2005, p. 89-96.

[7] David N BARTON, Sakari KUIKKA, Olli VARIS, Laura UUSITALO, Hans Jørgen HENRIKSEN, Mark BORSUK, Africa de la HERA, Raziyeh FARMANI, Sandra JOHNSON et John DC LINNELL, « Bayesian networks in environmental and resource management », in : *Integrated Environmental Assessment and Management* 8.*3* (2012), p. 418-429, DOI : https://doi.org/10.1002/ieam.1327, eprint : https:

//setac.onlinelibrary.wiley.com/doi/pdf/10.1002/ieam.1327, URL : https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/ieam.1327.

[8] R. Battiti, « Using mutual information for selecting features in supervised neural net learning », in : *IEEE Transactions on Neural Networks* 5.*4* (1994), p. 537-550, DOI : 10.1109/72.298224.

[9] Salem Benferhat, Philippe Leray et Karim Tabia, « Belief Graphical Models for Uncertainty Representation and Reasoning, in A Guided Tour of Artificial Intelligence Research : Volume II : AI Algorithms », in : Springer, 2020, p. 209-246, ISBN : 978-3-030-06167-8, DOI : 10.1007/978-3-030-06167-8_8, URL : https://doi.org/10.1007/978-3-030-06167-8_8.

[10] Tomas Beuzen, Lucy Marshall et Kristen D Splinter, « A comparison of methods for discretizing continuous variables in Bayesian Networks », in : *Environmental modelling & software* 108 (2018), p. 61-66.

[11] Yash Bhalgat, Zhe Liu, Pritam Gundecha, Jalal Mahmud et Amita Misra, « Teacher-student learning paradigm for tri-training : an efficient method for unlabeled data exploitation », in : *arXiv preprint arXiv :1909.11233* (2019).

[12] Concha Bielza et Pedro Larrañaga, « Discrete Bayesian Network Classifiers : A Survey », in : *ACM Comput. Surv.* 47.*1* (juill. 2014), ISSN : 0360-0300, DOI : 10.1145/2576868, URL : https://doi.org/10.1145/2576868.

[13] Avrim Blum et Tom Mitchell, « Combining labeled and unlabeled data with co-training », in : *COLT' 98 : Proceedings of the eleventh annual conference on Computational learning theory*, 1998, p. 92-100.

[14] Avrim L Blum et Pat Langley, « Selection of relevant features and examples in machine learning », in : *Artificial intelligence* 97.*1-2* (1997), p. 245-271.

[15] Wray Buntine, « Theory refinement on Bayesian networks », in : *Uncertainty proceedings 1991*, Elsevier, 1991, p. 52-60.

[16] Eugene Charniak, « Bayesian Networks without Tears. », in : *AI Magazine* 12.*4* (1991), p. 50, DOI : 10.1609/aimag.v12i4.918, URL : https://ojs.aaai.org/index.php/aimagazine/ARTICLE/view/918.

[17] Qiongyu Chen, Guoliang Li, Tze-Yun Leong et Chew-Kiat Heng, « Predicting coronary artery disease with medical profile and gene polymorphisms data », in : (2007).

[18] Jie Cheng et Russell Greiner, « Comparing Bayesian network classifiers », in : *arXiv preprint arXiv :1301.6684* (2013).

[19] Stephen Clark, James R Curran et Miles Osborne, « Bootstrapping POS-taggers using unlabelled data », in : *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, p. 49-55.

[20] Michael Collins et Yoram Singer, « Unsupervised models for named entity classification », in : *1999 Joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999.

[21] Diego Colombo, Marloes H Maathuis et al., « Order-independent constraint-based causal structure learning. », in : *J. Mach. Learn. Res.* 15.*1* (2014), p. 3741-3782.

[22] Gregory F Cooper et Edward Herskovits, « A Bayesian method for the induction of probabilistic networks from data », in : *Machine learning* 9.*4* (1992), p. 309-347.

[23] M. Correa, C. Bielza et J. Pamies-Teixeira, « Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process », in : *Expert Systems with Applications* 36.*3, Part 2* (2009), p. 7270-7279, issn : 0957-4174, doi : https://doi.org/10.1016/j.eswa.2008.09.024, url : https://www.sciencedirect.com/science/article/pii/S0957417408006593.

[24] M. Craig, T.J. Harvey, R.J.K. Wood, K. Masuda, M. Kawabata et H.E.G. Powrie, « Advanced condition monitoring of tapered roller bearings, Part 1 », English, in : *Tribology International* 42.*11-12* (2009), p. 1846-1856, doi : 10.1016/j.triboint.2009.04.033.

[25] Rónán Daly, Qiang Shen et Stuart Aitken, « Learning Bayesian networks : approaches and issues », in : *The knowledge engineering review* 26.*2* (2011), p. 99-157.

[26] Sanjoy Dasgupta, Michael Littman et David McAllester, « PAC generalization bounds for co-training », in : *Advances in neural information processing systems* 14 (2001).

[27] C. de Castelbajac, M. Ritou, S. Laporte et B. Furet, « Monitoring of distributed defects on HSM spindle bearings », in : *Applied Acoustics* 77 (2014), p. 159-168, ISSN : 0003-682X, DOI : https://doi.org/10.1016/j.apacoust.2013.07.008, URL : https://www.sciencedirect.com/science/ARTICLE/pii/S0003682X1300159X.

[28] A. P. Dempster, N. M. Laird et D. B. Rubin, « Maximum likelihood from incomplete data via the EM algorithm », in : *J. of the Royal Statistical Society : Series B* 39 (1977), p. 1-38.

[29] S. Dey et J.A. Stori, « A Bayesian network approach to root cause diagnosis of process variations », in : *International Journal of Machine Tools and Manufacture* 45.*1* (2005), p. 75-91, ISSN : 0890-6955, DOI : https://doi.org/10.1016/j.ijmachtools.2004.06.018, URL : https://www.sciencedirect.com/science/article/pii/S0890695504001609.

[30] Francisco Javier Diez, « Parameter adjustment in Bayes networks. The generalized noisy OR–gate », in : *Uncertainty in artificial intelligence*, Elsevier, 1993, p. 99-105.

[31] F Javier Diez et Marek J Druzdzel, « Canonical probabilistic models for knowledge engineering », in : *UNED, Madrid, Spain, Technical Report CISIAD-06-01* (2006).

[32] James Dougherty, Ron Kohavi et Mehran Sahami, « Supervised and Unsupervised Discretization of Continuous Features », in : *Machine Learning Proceedings 1995*, sous la dir. d'Armand Prieditis et Stuart Russell, San Francisco (CA) : Morgan Kaufmann, 1995, p. 194-202, ISBN : 978-1-55860-377-6, DOI : https://doi.org/10.1016/B978-1-55860-377-6.50032-3, URL : https://www.sciencedirect.com/science/article/pii/B9781558603776500323.

[33] Dheeru Dua et Casey Graff, *UCI Machine Learning Repository*, 2017, URL : http://archive.ics.uci.edu/ml.

[34] Stephan Ebersbach et Zhongxiao Peng, « Expert system development for vibration analysis in machine condition monitoring », English, in : *Expert Systems With Applications* 34.*1* (2008), p. 291-299, DOI : 10.1016/j.eswa.2006.09.029.

[35] Raziyeh Farmani, Hans Jørgen Henriksen, Dragan Savic et David Butler, « An evolutionary Bayesian belief network methodology for participatory decision making under uncertainty : An application to groundwater management », in : *In-*

tegrated Environmental Assessment and Management 8.*3* (2012), p. 456-461, DOI : https://doi.org/10.1002/ieam.192, eprint : https://setac.onlinelibrary.wiley.com/doi/pdf/10.1002/ieam.192, URL : https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/ieam.192.

[36] Usama FAYYAD et Keki IRANI, « Multi-interval discretization of continuous-valued attributes for classification learning », in : (1993).

[37] Felix FEGER et Irena KOPRINSKA, « Co-training using rbf nets and different feature splits », in : *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, IEEE, 2006, p. 1878-1885.

[38] Michael N. FIENEN et Nathaniel G. PLANT, « A cross-validation package driving Netica with python », in : *Environmental Modelling & Software* 63 (2015), p. 14-23, ISSN : 1364-8152, DOI : https://doi.org/10.1016/j.envsoft.2014.09.007, URL : https://www.sciencedirect.com/science/article/pii/S1364815214002606.

[39] Nir FRIEDMAN, « The Bayesian structural EM algorithm », in : *arXiv preprint arXiv :1301.7373* (2013).

[40] R. GAO, L. WANG, R. TETI, D. DORNFELD, S. KUMARA, M. MORI et M. HELU, « Cloud-enabled prognosis for manufacturing », English, in : *CIRP Annals - Manufacturing Technology* 64.*2* (2015), p. 749-772.

[41] Victor GODREAU, Mathieu RITOU, Etienne CHOVÉ, Benoît FURET et Didier DUMUR, « Continuous improvement of HSM process by data mining », in : *J. of Int. Manufacturing* 30.*7* (oct. 2019), p. 2781-2788, DOI : 10.1007/s10845-018-1426-7, URL : https://hal.archives-ouvertes.fr/hal-01819020.

[42] Sally GOLDMAN et Yan ZHOU, « Enhancing supervised learning with unlabeled data », in : *ICML*, 2000, p. 327-334.

[43] Sam GRIFFITHS-JONES, « miRBase : the microRNA sequence database », in : *MicroRNA Protocols* (2006), p. 129-138.

[44] Daniel GROSSMAN et Pedro DOMINGOS, « Learning Bayesian network classifiers by maximizing conditional likelihood », in : *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 46.

[45] Asela Gunawardana et Chris Meek, « Universal Models of Multivariate Temporal Point Processes », in : *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, sous la dir. d'Arthur Gretton et Christian C. Robert, t. 51, Proceedings of Machine Learning Research, Cadiz, Spain : PMLR, sept. 2016, p. 556-563, URL : https://proceedings.mlr.press/v51/gunawardana16.html.

[46] Xiangyu Guo et Wei Wang, « Towards making co-training suffer less from insufficient views », in : *Frontiers of Computer Science* 13.*1* (fév. 2019), p. 99-105.

[47] Isabelle Guyon et André Elisseeff, « An introduction to variable and feature selection », in : *Journal of machine learning research* 3.*Mar* (2003), p. 1157-1182.

[48] Ümit Güz, Sébastien Cuendet, Dilek Hakkani-Tür et Gökhan Tür, « Co-training using prosodic and lexical information for sentence segmentation. », in : *Interspeech*, 2007, p. 2597-2600.

[49] Tengda Han, Weidi Xie et Andrew Zisserman, « Self-supervised co-training for video representation learning », in : *Advances in Neural Information Processing Systems* 33 (2020), p. 5679-5690.

[50] Anca Hanea, Oswaldo Morales Napoles et Dan Ababei, « Non-parametric Bayesian networks : Improving theory and reviewing applications », in : *Reliability Engineering & System Safety* 144 (2015), p. 265-284, ISSN : 0951-8320, DOI : https://doi.org/10.1016/j.ress.2015.07.027, URL : https://www.sciencedirect.com/science/article/pii/S0951832015002331.

[51] David Heckerman et Dan Geiger, « Likelihoods and parameter priors for Bayesian networks », in : *arXiv preprint arXiv :2105.06241* (2021).

[52] David Heckerman, Dan Geiger et David M Chickering, « Learning Bayesian networks : The combination of knowledge and statistical data », in : *Machine learning* 20.*3* (1995), p. 197-243.

[53] Max Henrion, « Some Practical Issues in Constructing Belief Networks. », in : *UAI*, t. 3, 1987, p. 161-173.

[54] Nazrul Hoque, Dhruba K Bhattacharyya et Jugal K Kalita, « MIFS-ND : A mutual information-based feature selection method », in : *Expert Systems with Applications* 41.*14* (2014), p. 6371-6385.

[55] Hui-Huang Hsu, Cheng-Wei Hsieh et Ming-Da Lu, « Hybrid feature selection by combining filters and wrappers », in : *Expert Systems with Applications* 38.*7* (2011), p. 8144-8150.

[56] PoTsang B. Huang, Cheng-Chieh Ma et Chia-Hao Kuo, « A PNN self-learning tool breakage detection system in end milling operations », in : *Applied Soft Computing* 37 (2015), p. 114-124.

[57] Rebecca Hwa, Miles Osborne, Anoop Sarkar et Mark Steedman, « Corrected co-training for statistical parsers », in : *Working Notes of the ICML'03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Citeseer, 2003.

[58] Abdal Moughit Idrissi Badssi, « Probabilistic graphical models for machining monitoring », mém. de mast., Université de Nantes, rapport de stage de Master, 2017.

[59] ISO, *ISO 10816-3 :2009 Mechanical vibration - Evaluation of machine vibration by measurements on non-rotating parts*, standard, Int. Org. for Standardization, 2009.

[60] ISO, *ISO 17243-1 :2014 Machine tool spindles - Evaluation of machine tool spindle vibrations by measurements on spindle housing*, standard, Int. Org. for Standardization, 2014.

[61] Aida Jarraya, Philippe Leray et Afif Masmoudi, « Discrete exponential Bayesian networks : definition, learning and application for density estimation », in : *Neurocomputing* 137 (2014), p. 142-149.

[62] Zhiwei Ji, Qibiao Xia et Guanmin Meng, « A review of parameter learning methods in Bayesian network », in : *International Conference on Intelligent Computing*, Springer, 2015, p. 3-12.

[63] Sheng-yi Jiang, Xia Li, Qi Zheng et Lian-xi Wang, « Approximate Equal Frequency Discretization Method », in : *intervals* 1.*1* ().

[64] Alberto Jimenez-Cortadi, Itziar Irigoien, Fernando Boto, Basilio Sierra et German Rodriguez, « Predictive Maintenance on the Machining Process and Machine Tool », in : *Applied Sciences* 10.*1* (2020), p. 1-14, issn : 2076-3417, url : https://www.mdpi.com/2076-3417/10/1/224.

[65] George H John, Ron Kohavi et Karl Pfleger, « Irrelevant features and the subset selection problem », in : *Machine learning proceedings 1994*, Elsevier, 1994, p. 121-129.

[66] Sandra Johnson et Kerrie Mengersen, « Integrated Bayesian network framework for modeling complex ecological issues », in : *Integrated Environmental Assessment and Management* 8.*3* (2012), p. 480-490, doi : https://doi.org/10.1002/ieam.274, eprint : https://setac.onlinelibrary.wiley.com/doi/pdf/10.1002/ieam.274, url : https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/ieam.274.

[67] Yan Ju, Lingling Li, Licheng Jiao, Zhongle Ren, Biao Hou et Shuyuan Yang, « Modified diversity of class probability estimation co-training for hyperspectral image classification », in : *arXiv preprint arXiv :1809.01436* (2018).

[68] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev et al., « The kinetics human action video dataset », in : *arXiv preprint arXiv :1705.06950* (2017).

[69] Dong-Hyeon Kim, Thomas J. Y. Kim, Xinlin Wang, Mincheol Kim, Ying-Jun Quan, Jin Woo Oh, Soo-Hong Min, Hyungjung Kim, Binayak Bhandari, Insoon Yang et Sung-Hoon Ahn, « Smart Machining Process Using Machine Learning : A Review and Perspective on Machining Industry », in : *Int. J. of Precis. Eng. and Manuf.-Green Tech.* 5 (2018), p. 555-568.

[70] Svetlana Kiritchenko et Stan Matwin, « Email classification with co-training », in : *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, Citeseer, 2001, p. 8.

[71] William Klement, Szymon Wilk, Wojtek Michalowski, Ken J Farion, Martin H Osmond et Vedat Verter, « Predicting the need for CT imaging in children with minor head injury using an ensemble of Naive Bayes classifiers », in : *Artificial intelligence in medicine* 54.*3* (2012), p. 163-170.

[72] Igor Kononenko, « On biases in estimating multi-valued attributes », in : *Ijcai*, t. 95, Citeseer, 1995, p. 1034-1040.

[73] Hildegard Kuehne, Hueihan Jhuang, Estıébaliz Garrote, Tomaso Poggio et Thomas Serre, « HMDB : a large video database for human motion recognition », in : *2011 International conference on computer vision*, IEEE, 2011, p. 2556-2563.

[74] E. Kuljanic, G. Totis et M. Sortino, « Development of an intelligent multisensor chatter detection system in milling », English, in : *Mechanical Systems and Signal Processing* 23.*5* (2009), p. 1704-1718, doi : 10.1016/j.ymssp.2009.01.003.

[75] Nojun Kwak et Chong-Ho Choi, « Input feature selection by mutual information based on Parzen window », in : *IEEE transactions on pattern analysis and machine intelligence* 24.*12* (2002), p. 1667-1671.

[76] Mostafa Langarizadeh et Fateme Moghbeli, « Applying naive bayesian networks to disease prediction : a systematic review », in : *Acta Informatica Medica* 24.*5* (2016), p. 364.

[77] Pat Langley, Wayne Iba, Kevin Thompson et al., « An analysis of Bayesian classifiers », in : *Aaai*, t. 90, 1992, p. 223-228.

[78] Guillaume Lemaître, Fernando Nogueira et Christos K. Aridas, « Imbalanced-learn : A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning », in : *Journal of Machine Learning Research* 18.*17* (2017), p. 1-5, url : http://jmlr.org/papers/v18/16-365.

[79] Ming Li et Zhi-Hua Zhou, « Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples », in : *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans* 37.*6* (2007), p. 1088-1098.

[80] Xiaoyue Li, Yinfei Yang, Liang Li, Guolong Zhao et Ning He, « Uncertainty quantification in machining deformation based on Bayesian network », in : *Reliability Engineering & System Safety* 203 (2020), p. 107113, issn : 0951-8320, doi : https://doi.org/10.1016/j.ress.2020.107113, url : https://www.sciencedirect.com/science/article/pii/S0951832020306141.

[81] Charles X. Ling, Jun Du et Zhi-Hua Zhou, « When does Co-training Work in Real Data ? », in : *Advances in Knowledge Discovery and Data Mining*, sous la dir. de Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone et Tu-Bao Ho, Springer Berlin Heidelberg, 2009, p. 596-603.

[82] Ahmed Mabrouk, Christophe Gonzales, Karine Jabet-Chevalier et Eric Chojnaki, « Multivariate Cluster-Based Discretization for Bayesian Network Structure Learning », in : *Proc. of SUM'15*, 2015, p. 155-169.

[83] J. Mathew, « The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis », in : *Journal of Vibration, Acoustics, Stress, and Reliability in Design* 106.*6* (1984), p. 447-453, URL : https://ci.nii.ac.jp/naid/80002742403/en/.

[84] Edson T Matsubara, Maria C Monard et Ronaldo C Prati, « On the class distribution labelling step sensitivity of CO-TRAINING », in : *IFIP International Conference on Artificial Intelligence in Theory and Practice*, Springer, 2006, p. 199-208.

[85] Scott McLachlan, Kudakwashe Dube, Graham A Hitman, Norman E Fenton et Evangelia Kyrimi, « Bayesian networks in healthcare : Distribution by medical condition », in : *Artificial Intelligence in Medicine* 107 (2020), p. 101912, ISSN : 0933-3657, DOI : https://doi.org/10.1016/j.artmed.2020.101912, URL : https://www.sciencedirect.com/science/article/pii/S0933365720300774.

[86] Christopher Meek, « Toward Learning Graphical and Causal Process Models. », in : *CI@ UAI*, 2014, p. 43-48.

[87] John Mingers, « An empirical comparison of pruning methods for decision tree induction », in : *Machine learning* 4.*2* (1989), p. 227-243.

[88] Mathilde Monvoisin, Philippe Leray et Mathieu Ritou, « Unsupervised co-training of Bayesian networks for condition prediction », in : *Proc. of IEA/AIE 2021*, 2021.

[89] Mathilde Monvoisin, Philippe Leray et Mathieu Ritou, « Unsupervised condition monitoring with bayesian networks : an application on high speed machining », in : *31th European Safety and Reliability Conference, ESREL 2021*, 2021, p. 1990-1997.

[90] Dinora A Morales, Yolanda Vives-Gilabert, Beatriz Gómez-Ansón, Endika Bengoetxea, Pedro Larrañaga, Concha Bielza, Javier Pagonabarraga, Jaime Kulisevsky, Idoia Corcuera-Solano et Manuel Delfino, « Predicting dementia development in Parkinson's disease using Bayesian network classifiers », in : *Psychiatry Research : NeuroImaging* 213.*2* (2013), p. 92-98.

[91] Ali Ben Mrad, Véronique Delcroix, Sylvain Piechowiak, Philip Leicester et Mohamed Abid, « An explication of uncertain evidence in Bayesian networks : likelihood evidence and probabilistic evidence », in : *Applied Intelligence* 43.*4* (2015), p. 802-824.

[92] J. Munoa, X. Beudaert, Z. Dombovari, Y. Altintas, E. Budak, C. Brecher et G. Stepan, « Chatter suppression techniques in metal cutting », in : *CIRP Annals* 65.*2* (2016), p. 785-808, issn : 0007-8506, doi : https://doi.org/10.1016/j.cirp.2016.06.004, url : https://www.sciencedirect.com/science/ARTICLE/pii/S0007850616301962.

[93] K.P. Murphy, « Dynamic Bayesian Networks : Representation, Inference and Learning », thèse de doct., Berkeley : University of california, 2002, url : http://www.ai.mit.edu/~murphyk/Thesis/thesis.html.

[94] Ion Muslea, Steven Minton et Craig A Knoblock, « Selective sampling with redundant views », in : *AAAI/IAAI*, 2000, p. 621-626.

[95] Dinh Son Nguyen, « Application of Bayesian networks for product quality management in a multistage manufacturing process », in : *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE, 2015, p. 1402-1406.

[96] Kamal Nigam et Rayid Ghani, « Analyzing the Effectiveness and Applicability of Co-Training », in : *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM 00, New York, NY, USA : Association for Computing Machinery, 2000, p. 86-93.

[97] Xin Ning, Xinran Wang, Shaohui Xu, Weiwei Cai, Liping Zhang, Lina Yu et Wenfa Li, « A review of research on co-training », in : *Concurrency and Computation : Practice and Experience* n/a.*n/a* (), e6276, doi : https://doi.org/10.1002/cpe.6276, eprint : https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6276, url : https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6276.

[98] Xuesong Niu, Hu Han, Shiguang Shan et Xilin Chen, « Multi-label co-regularization for semi-supervised facial action unit recognition », in : *Advances in neural information processing systems* 32 (2019).

[99] Farnaz Nojavan, Song S Qian et Craig A Stow, « Comparative analysis of discretization methods in Bayesian networks », in : *Environmental Modelling & Software* 87 (2017), p. 64-71.

[100] Agnieszka Oniśko, Marek J Druzdzel et Hanna Wasyluk, « Learning Bayesian network parameters from small data sets : Application of Noisy-OR gates », in : *International Journal of Approximate Reasoning* 27.*2* (2001), p. 165-182.

[101] Judea Pearl, *Bayesian networks, causal inference and knowledge discovery*, rapp. tech., UCLA Cognitive Systems Laboratory, Technical Report, 2001.

[102] Judea Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*, Morgan kaufmann, 1988.

[103] Grégoire Peigne, Henri Paris, Daniel Brissaud et Alexandre Gouskov, « Impact of the cutting dynamics of small radial immersion milling operations on machined surface roughness », English, in : *Int. J. of Machine Tools and Manufacture* 44.*11* (2004), p. 1133-1142, doi : 10.1016/j.ijmachtools.2004.04.012.

[104] David Pierce et Claire Cardie, « Limitations of co-training for natural language learning from large datasets », in : *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001.

[105] Nathaniel G. Plant et K. Todd Holland, « Prediction and assimilation of surf-zone processes using a Bayesian network : Part I : Forward models », in : *Coastal Engineering* 58.*1* (2011), p. 119-130, issn : 0378-3839, doi : https://doi.org/10.1016/j.coastaleng.2010.09.003, url : https://www.sciencedirect.com/science/article/pii/S0378383910001353.

[106] Carmel A. Pollino, Owen Woodberry, Ann Nicholson, Kevin Korb et Barry T. Hart, « Parameterisation and evaluation of a Bayesian network for use in an ecological risk assessment », in : *Environmental Modelling & Software* 22.*8* (2007), Bayesian networks in water resource modelling and management, p. 1140-1152, issn : 1364-8152, doi : https://doi.org/10.1016/j.envsoft.2006.03.006, url : https://www.sciencedirect.com/science/article/pii/S1364815206000788.

[107] R. Ramesh, M.A. Mannan, A.N. Poo et S.S. Keerthi, « Thermal error measurement and modelling in machine tools. Part II. Hybrid Bayesian Network—support

vector machine model », in : *International Journal of Machine Tools and Manufacture* 43.*4* (2003), p. 405-419.

[108] Robert B. Randall et Jérôme Antoni, « Rolling element bearing diagnostics—A tutorial », English, in : *Mechanical Syst. and Signal Proc.* 25.*2* (2011), p. 485-520, issn : 0888-3270, doi : 10.1016/j.ymssp.2010.07.017.

[109] Jiangtao Ren, Sau Dan Lee, Xianlu Chen, Ben Kao, Reynold Cheng et David Cheung, « Naive bayes classification of uncertain data », in : *2009 Ninth IEEE international conference on data mining*, IEEE, 2009, p. 944-949.

[110] Kurt S Riedel, *Detection of abrupt changes : theory and application*, 1994.

[111] Devni Prima Sari, Dedi Rosadi, Adhitya Ronnie Effendie et Danardono Danardono, « Discretization methods for Bayesian networks in the case of the earthquake », in : *Bulletin of Electrical Engineering and Informatics* 10.*1* (2021), p. 299-307.

[112] Ville Satopaa, Jeannie Albrecht, David Irwin et Barath Raghavan, « Finding a" kneedle" in a haystack : Detecting knee points in system behavior », in : *2011 31st international conference on distributed computing systems workshops*, IEEE, 2011, p. 166-171.

[113] Mohsen Sheikh Hassani et James R Green, « Multi-view Co-training for microRNA Prediction », in : *Scientific reports* 9.*1* (2019), p. 1-10.

[114] Dan Song, Carl Henrik Ek, Kai Huebner et Danica Kragic, « Multivariate discretization for bayesian network structure learning in robot grasping », in : *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, p. 1944-1950.

[115] Khurram Soomro, Amir Roshan Zamir et Mubarak Shah, « A dataset of 101 human action classes from videos in the wild », in : *Center for Research in Computer Vision* 2.*11* (2012).

[116] Peter Spirtes, Clark N Glymour, Richard Scheines et David Heckerman, *Causation, prediction, and search*, MIT press, 2000.

[117] Masami Takikawa et Bruce D'Ambrosio, « Multiplicative factorization of noisy-max », in : *arXiv preprint arXiv :1301.6742* (2013).

[118] HL Tang, ZK Lin, MY Lu et al., « An advanced co-training algorithm based on mutual independence and diversity measures », in : *Journal of Computer Research and Development* 45.*11* (2008), p. 1874-1881.

[119] Fei Tao, Qinglin Qi, Ang Liu et Andrew Kusiak, « Data-driven smart manufacturing », in : *J. of Manufacturing Systems* 48 (2018), Special Issue on Smart Manufacturing, p. 157-169.

[120] R. Teti, K. Jemielniak, G. O'Donnell et D. Dornfeld, « Advanced monitoring of machining operations », in : *CIRP Annals* 59.*2* (2010), p. 717-739, issn : 0007-8506, doi : https://doi.org/10.1016/j.cirp.2010.05.010, url : https://www.sciencedirect.com/science/ARTICLE/pii/S0007850610001976.

[121] Khaoula Tidriri, Nizar Chatti, Sylvain Verron et Teodor Tiplica, « Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring : A review of researches and future challenges », English, in : *Annual Reviews in Control* 42.*C* (2016), p. 63-81, doi : 10.1016/j.arcontrol.2016.09.008.

[122] Xavier Tolsa, « Principal values for the Cauchy integral and rectifiability », in : *Proceedings of the American Mathematical Society* 128.*7* (2000), p. 2111-2119.

[123] Gokhan Tur, « Co-adaptation : Adaptive co-training for semi-supervised learning », in : *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, p. 3721-3724.

[124] Laura Uusitalo, « Advantages and challenges of Bayesian networks in environmental modelling », in : *Ecological Modelling* 203.*3* (2007), p. 312-318, issn : 0304-3800, doi : https://doi.org/10.1016/j.ecolmodel.2006.11.033, url : https://www.sciencedirect.com/science/ARTICLE/pii/S0304380006006089.

[125] Laura Uusitalo, Sakari Kuikka, Pirkko Kauppila, Pirkko Söderkultalahti et Saara Bäck, « Assessing the roles of environmental factors in coastal fish production in the northern Baltic Sea : A Bayesian network application », in : *Integrated environmental assessment and management* 8.*3* (2012), p. 445-455.

[126] S. Vafaei, H. Rahnejat et R. Aini, « Vibration monitoring of high speed spindles using spectral analysis techniques », English, in : *Int. J. of Machine Tools and Manufacture* 42.*11* (2002), p. 1223-1234.

[127] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N Kavuri et Kewen Yin, « A review of process fault detection and diagnosis : Part III : Process history based methods », in : *Computers & chemical engineering* 27.*3* (2003), p. 327-346.

[128] Pauli Virtanen et al., « SciPy 1.0 : Fundamental Algorithms for Scientific Computing in Python », in : *Nature Methods* 17 (2020), p. 261-272, doi : 10.1038/s41592-019-0686-2.

[129] J Vomlel, « Probabilistic reasoning with uncertain evidence », in : *Neural network world* 14 (2004), p. 453-466.

[130] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X. Gao et Dazhong Wu, « Deep learning for smart manufacturing : Methods and applications », in : *Journal of Manufacturing Systems* 48 (2018), Special Issue on Smart Manufacturing, p. 144-156.

[131] Kung-Jeng Wang, Bunjira Makond et Kung-Min Wang, « Modeling and predicting the occurrence of brain metastasis from lung cancer by Bayesian network : a case study of Taiwan », in : *Computers in biology and medicine* 47 (2014), p. 147-160.

[132] Mingwei Wang et Jingtao Zhou, « A Bayesian network-based classifier for machining error prediction », in : *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, 2014, p. 841-844.

[133] Wei Wang et Zhi-Hua Zhou, « Co-Training with Insufficient Views », in : *Proceedings of the 5th Asian Conference on Machine Learning*, sous la dir. de Cheng Soon Ong et Tu Bao Ho, t. 29, Proceedings of Machine Learning Research, Australian National University, Canberra, Australia : PMLR, nov. 2013, p. 467-482, url : https://proceedings.mlr.press/v29/Wang13b.html.

[134] Wei Wang et Zhi-Hua Zhou, « On multi-view active learning and the combination with semi-supervised learning », in : *Proceedings of the 25th international conference on Machine learning*, 2008, p. 1152-1159.

[135] P. Weber, G. Medina-Oliva, C. Simon et B. Iung, « Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas », English, in : *Eng. App. of AI* 25.4 (2012), p. 671-682, doi : 10.1016/j.engappai.2010.06.002.

[136] David Yarowsky, « Unsupervised word sense disambiguation rivaling supervised methods », in : *33rd annual meeting of the association for computational linguistics*, 1995, p. 189-196.

[137] Shipeng Yu, Balaji Krishnapuram, Rómer Rosales et R. Bharat Rao, « Bayesian Co-Training », in : *Journal of Machine Learning Research* 12.*80* (2011), p. 2649-2680.

[138] Adam Zagorecki et Marek J Druzdzel, « Knowledge engineering for Bayesian networks : How common are noisy-MAX distributions in practice ? », in : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* 43.*1* (2012), p. 186-195.

[139] Qinpei Zhao, Ville Hautamaki et Pasi Fränti, « Knee point detection in BIC for detecting the number of clusters », in : *International conference on advanced concepts for intelligent vision systems*, Springer, 2008, p. 664-673.

[140] Zhi-Hua Zhou, Ke-Jia Chen et Hong-Bin Dai, « Enhancing relevance feedback in image retrieval using unlabeled data », in : *ACM Transactions on Information Systems (TOIS)* 24.*2* (2006), p. 219-244.

[141] Zhi-Hua Zhou, Ming Li et al., « Semi-supervised regression with co-training. », in : *IJCAI*, t. 5, 2005, p. 908-913.

[142] Zhi-Hua Zhou et Ming Li, « Tri-training : Exploiting unlabeled data using three classifiers », in : *IEEE Transactions on knowledge and Data Engineering* 17.*11* (2005), p. 1529-1541.

ANNEXE

Simulation des données

L'objectif de la création des données est de simuler des capteurs dont les signaux sont enregistrés en continu, et agrégés selon certains seuils de criticité tels que décrits dans l'équation 3.1 (voir sous-section 1.4.4), ainsi que les indicateurs de suivi de l'état associés.

Cette partie concerne la création des données de la vue A, qui simulent des données de critères calculés à partir de capteurs, et des données de la vue B, qui simulent un ensemble d'indicateurs de l'état.

Algorithme de génération des données artificielles

L'objectif de la création des données est de simuler des capteurs dont les signaux sont enregistrés en continu, et agrégés selon certains seuils de criticité tels que décrits dans l'équation 3.1 (voir sous-section 1.4.4), ainsi que les indicateurs de suivi de l'état associés. La variable correspondant au $v$ème seuil du $p$ème critère est désignée par $X^A_{p,v}$. Les données de chaque vue auront une taille N correspondant au nombre de jours d'enregistrement et N_j est le nombre de valeurs enregistrées chaque jour pour chaque capteur. Les niveaux normaux et critiques sont rassemblés dans une matrice de référence R pour chaque variable.

A chaque instance des données est associé un niveau ω, avec $\omega \in \mathbb{N}$ qui a une probabilité $P(\omega)$ de se produire. Si $\omega = 0$, cela signifie une absence d'événement. Si $\omega > 0$, alors une variable dépasse la valeur qui lui est considérée comme normale et un événement E se produit, $P(E) = P(\omega > 0)$. Un événement est associé à une durée T, et est considéré comme un événement long s'il dure au moins m mesures d'affilée. La probabilité d'obtenir un événement long est de $P(T > m)$. Une valeur générée pendant un événement long aura la même valeur sur toute la durée T, à $\sigma\%$ près, avec $\sigma \in [-1, 1]$ tiré aléatoirement à chaque itération. Les données A sont générées en suivant l'algorithme 9.

La création des données B se fait ensuite à partir des données A, pour simuler le fait que les événements de A sont ce qui provoquent des dégradations de l'état. La procédure est décrite dans l'algorithme 10. Pour cela, il faut définir l'impact qu'aura un événement

Algorithme 9 : Génération des données A

Entrée : Matrices des valeurs de références $\mathbf{R}^A$, m, N_j, N, $P(\omega)$, $P(T)$
Sortie : $\mathcal{X}^A$

```
1  événement long = faux
2  pour chaque critère X_p faire
3      pour chaque jour k de 1 à N faire
4          x^k_{p,v} = 0
5          pour chaque tirage j de 1 à N_j faire
6              si événement long alors
7                  T = T − 1
8                  μ = aléatoire uniforme(−1/100, 1/100)
9                  x^j_p = (1 + μ) x^{j−1}_p
10                 si T = 0 alors
11                     événement long = faux
12             sinon
13                 ω = aléatoire pondéré par les probabilités P(ω)
14                 x^j_p = aléatoire uniforme(R^A_{p,ω}, R^A_{p,ω+1})
15                 si ω > 0 alors
16                     événement long = aléatoire pondéré par les probabilités P(T)
17             pour chaque valeur de seuil τ_{v,p} faire
                   // formule 3.1
18                 si x^j_p > R^A_{p,v} alors
19                     x^k_{p,v} = x^k_{p,v} + x^j_p
20         ajouter x^k_{p,v} à x_{p,v}
```

observé dans les données A sur les données B. Il est donc défini pour chaque indicateur, lesquels des critères et des seuils aura un impact plus ou moins fort sur la valeur des indicateurs. Ces éléments sont appelés éléments impactant z, et sont exprimés sous la forme de triplet tel que dans l'équation 5.1, avec un critère X_p, un seuil $\tau_{v,p}$ et un niveau ω.

$$z = (X_p, \tau_{v,p}, \omega) \tag{5.1}$$

La fonction $\gamma_k(z)$ de l'équation 5.2 donne la valeur du coefficient multiplicateur pour chaque instance k de la vue B. Le vecteur de valeurs du critère $X_{p,v}$ est noté $\mathbf{x}_{p,v}$, et $x^k_{p,v}$ est sa $k$ème valeur, c'est-à-dire sa valeur au $k$ème jour. De la même façon, le vecteur de valeurs de la l'indicateur X_q est noté $\mathbf{x}_q$, et x^k_q est sa $k$ème valeur.

$$\gamma_k(z) = \frac{x^k_{p,v}}{max(\mathbf{x}_{p,v})} \tag{5.2}$$

Algorithme 10 : Génération des données B

Entrée : Matrices des valeurs de références $\mathbf{R}^B$, ensemble des éléments impactant $\mathbf{z}$
Sortie : $\mathcal{X}^B$

```
pour chaque indicateur X_q faire
    pour chaque jour k de 1 à N faire
        x_q^k = 0
        pour chaque élément impactant z de l'indicateur X_q faire
            z = (c_p, s_v, ω)
            calcul de γ_k(z) // voir équation 5.2
            x_q^k = x_q^k + γ_k(z) × R^B_{q,ω}
        ajouter x_q^k à x_q
```

Enfin, le vecteur de vérité est généré à partir des données B, la valeur de vérité étant 0 quand tout se déroule normalement, et 1 lorsqu'un incident se produit, ce qui signifie qu'un des événements a déclenché une dégradation de l'état. La procédure de génération du vecteur de vérité se trouve dans l'algorithme 11. Il est considéré qu'une dégradation s'est produite lorsque la valeur de l'instance dépasse la moitié de la plus forte valeur pour une variable donnée.

Algorithme 11 : Génération de la vérité

Entrée : $\mathcal{X}^B$
Sortie : y

```
pour chaque jour k de 1 à N faire
    y^k = 0 // par défaut il n'y a pas d'incident
    pour chaque indicateur i_q faire
        si x_q^k > 1/2 max(x_q) alors
            y^k = 1
```

	min	s1	s2	s3	max
c_1	50	12500	25000	37500	50000
c_2	100	25000	50000	75000	100000
c_3	150	37500	75000	112500	150000
c_4	20	5000	10000	15000	20000

TABLE 5.5 – Matrice des valeurs de référence R^A de la vue $\mathcal{X}^A$ pour chaque critère.

Mise en application

En utilisant l'algorithme présenté précédemment, quatre jeux de données ont été générés, avec des valeurs d'impact distinctes mais avec les mêmes valeurs de référence, pour permettre de tirer des conclusions d'une comparaison de ces valeurs d'impact.

Dans cette exécution de l'algorithme, $m = 15$ et $\omega \in \{0, 1, 2, 3\}$. Les matrices des valeurs de référence R^A et R^B sont détaillées respectivement dans les tables 5.5 et 5.6. $N = 300$ et $N_j = 500$. La probabilité d'obtenir un événement long est $P(T > m) = 0.4$. La distribution de probabilités pour ω est : $P(\omega = 0) = 0.99832$, $P(\omega = 1) = 0.0008$, $P(\omega = 2) = 0.0008$, $P(\omega = 3) = 8e^{-5}$.

Le jeu de données 1 (jdd1) a des indicateurs très influencés par les seuils 0 et 2 des critères 1 et 3, mais également par les seuils 1 et 2 du critère 4, en moindre mesure.

Le jeu de données 2 (jdd2) fait plutôt ressortir les événements longs mais peu violents,

	OK	deg	KO
i1	60	1800	9000
i2	70	2100	10500
i3	20	600	3000
i4	90	2700	13500

TABLE 5.6 – Matrice des valeurs de référence R^B de la vue $\mathcal{X}^B$ pour chaque indicateur.

	i1	i2	i3	i4
jdd1	(c1,2,2)	(c1,0,1) (c2,0,1)	(c3,2,2)	(c4,1,1) (c4,2,1)
jdd2	(c1,0,2)	(c1,0,2) (c2,0,2)	(c3,0,2)	(c4,2,1)
jdd3	(c1,0,2) (c1,1,2) (c1,2,2)	(c2,2,1)	(c3,2,1)	(c4,2,1)
jdd4	(c1,0,2) (c2,0,2) (c3,0,2) (c4,0,2)	(c1,1,2) (c2,1,2) (c3,1,2) (c4,1,2)	(c1,1,1) (c2,1,1) (c3,1,1) (c4,1,1)	(c1,2,2) (c2,2,2) (c3,2,2) (c4,2,2)

TABLE 5.7 – Matrice des valeurs de référence R^B de la vue $\mathcal{X}^B$ pour chaque jeu de données.

surtout avec les critères 1, 2 et 3. Le critère 4 influe légèrement avec le seuil 2.

Le jeu de données 3 (jdd3), à l'inverse, est plus impacté par les événements courts et violents, essentiellement avec le critère 1 mais aussi les 3 autres, plus faiblement.

Enfin, le jeu de données 4 (jdd4) est le seul pour lequel tous les critères contribuent à part égales. Tous les seuils ont de l'importance, mais le seuil 1 influe légèrement plus fortement.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

Titre : Modèles graphiques probabilistes appliqués aux procédés de fabrication

Mot clés : Apprentissage, réseaux bayésiens, co-training, diagnostic, industrie 4.0

Résumé : La fabrication intelligente est un domaine de recherche prometteur pour l'amélioration de la productivité et de la compétitivité dans l'industrie, par l'exploitation des données numériques obtenues lors de procédés de fabrication, tel que l'usinage à grande vitesse. Les réseaux bayésiens ont fait leurs preuves en matière de classification et de diagnostic, et ils ont notamment l'intérêt d'être grandement interprétables. Cette thèse présente une architecture générique de réseaux bayésiens pour le diagnostic à partir de capteurs, incluant un mécanisme de sélection de variables basé sur l'information mutuelle. Le co-training est un champ émergent des algorithmes d'apprentissage à partir de données, et l'exploration de cette famille d'algorithmes est jusqu'à présent essentiellement limitée à un apprentissage supervisé ou semi-supervisé. Ce manuscrit propose plusieurs stratégies de co-training non-supervisées utilisables par tout modèle probabiliste, et détaille leur utilisation sur plusieurs jeux de données. L'ensemble des contributions théoriques est mis à profit dans un cas d'usage sur l'usinage à grande vitesse, dans lequel deux réseaux bayésiens avec la structure générique proposée permettent d'exploiter les données de capteurs d'une électrobroche en conditions réelles d'utilisation, et dont les paramètres sont appris grâce aux stratégies de co-training non-supervisées.

Title: Probabilistic graphical models applied to manufacturing processes

Keywords: Learning, Bayesian networks, co-training, diagnosis, industry 4.0

Abstract: Smart manufacturing is a promising area of research for improving productivity and competitiveness in industry, by exploiting digital data obtained during manufacturing processes, such as high-speed machining. Bayesian networks have proven their worth in classification and diagnosis, and they have the particular advantage of being highly interpretable. This thesis presents a generic Bayesian network architecture for sensor-based diagnosis, including a variable selection mechanism based on mutual information. Co-training is an emerging field of data-driven learning algorithms, and the exploration of this family of algorithms is so far mostly limited to supervised or semi-supervised learning. This manuscript proposes several unsupervised co-training strategies that can be used by any probabilistic model, and details their use on several datasets. All the theoretical contributions are put to use in a use case on high speed machining, in which two Bayesian networks with the proposed generic structure are used to exploit sensor data of an electrospindle in real conditions of use, and whose parameters are learned thanks to the unsupervised co-training strategies.