Contributions aux ombres et jumeaux numériques dans l'industrie : proposition d'une stratégie de couplage entre modèles de simulation et d'apprentissage automatique appliquée aux scieries

Sylvain Chabanet

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

THÈSE

présentée et soutenue publiquement le 21 Septembre 2023

pour l'obtention du grade de

Docteur de l'Université de Lorraine

(**mention Automatique**, Traitement du signal et des images, Génie informatique)

par

M. Sylvain Chabanet

Contributions aux ombres et jumeaux numériques dans l'industrie : proposition d'une stratégie de couplage entre modèles de simulation et d'apprentissage automatique appliquée aux scieries

Composition du jury

Président :	Samir Lamouri, Pr.	Arts et Métiers ParisTech ENSAM
Rapporteurs :	Jonathan Weber, Pr.	Université de Haute-Alsace
	Nathalie Julien, Pr	Université de Bretagne-Sud
Examinateurs :	Hind Bril El-Haouzi, Pr. (Dir.)	Université de Lorraine
	Philippe Thomas, MdC. (Co-Dir.)	Université de Lorraine
	Robert Pellerin, Pr.	École Polytechnique de Montréal

Centre de Recherche en Automatique de Nancy, CRAN UMR 7039, Université de Lorraine - CNRS

cnrs

CRAN

Mis en page avec la classe thesul.

Remerciements

Ce document constitue la conclusion d'un long projet, n'ayant pu être mené et aboutir dans de bonnes conditions que grâce à l'engagement de nombreuses personnes que je tiens à saluer ici.

Je remercie en premier lieu Hind Bril El-Haouzi et Philippe Thomas pour leur forte implication dans la direction de cette thèse, de l'écriture du sujet à la soutenance. C'était une chance d'avoir des encadrants aussi accessibles et disponibles, que ce soit pour échanger des idées et des pistes de recherche, relire et corriger documents et présentations, ou simplement discuter et s'aérer l'esprit. Pour tout cela, merci.

Evidemment, je tiens à remercier les membres du jury : Pr. Nathalie Julien et Pr. Jonathan Weber, pour avoir acccepté de rapporter cette thèse, ainsi que Pr. Samir Lamouri et Pr. Robert Pellerin pour avoir accepté d'en être examinateurs. Je suis heureux que vous ayez accepté de consacrer votre temps à l'évaluation de mes travaux et d'avoir pu bénéficier de vos différents retours.

Tout mes remerciements à Emmanuel Zimmermann, pour son implication et son soutient technique et moral à de nombreuses thèses du site du CRAN d'Epinal. Je remercie également Michael Morin et Jonathan Gaudreault pour nos différentes collaborations qui furent très agréables.

Et enfin, pêle-mêle, je remercie (et félicite !) ma famille, mes amis, et mes collègues, pour m'avoir supporter tout au long de ce projet. Discussions, soirées, jeux de société, vacances et voyages : toutes ces choses permettent de garder à l'esprit qu'il y a une vie en dehors du doctorat et qu'il est important de garder un équilibre entre la recherche et les loisirs.

Abstract

Toward industrial digital shadows and twins : a novel strategy to couple simulation and machine learning models, applied to the lumber industry

This thesis is part of the ANR project Lorraine-Artificial Intelligence, a multi-disciplinary project promoting research into both artificial intelligence itself, and its applications to other fields. As such, this thesis focuses on the development and use of machine learning models as a substitute for simulation models. Interest in this research topic is fueled by academic and industrial interest in the concept of digital shadows and twins, seen as an evolution of simulation models for long-term use at the heart of systems and processes.

The main contribution of this thesis is the proposal of a coupling strategy between a simulation model and a surrogate model performing the same prediction task repeatedly on a data stream. The simulation model is assumed to have a high level of fidelity, but to be too slow or computationally expensive to be used alone to perform the full range of prediction required. The surrogate model is a fast machine-learning model that approximates the simulation model. The primary objective of the proposed coupling strategy is the efficient use of limited computational resources by intelligently allocating each prediction request to one of the two models. This allocation is, in particular, inspired by active learning and based on the evaluation of the level of confidence in the predictions of the machine learning model. Numerical experiments are first carried out on eight datasets from the scientific literature. An application to the sawmilling industry is then developed.

Keywords: *Sawmill*, *Machine learning*, *Active learning*,*Digital Shadow*, *Surrogate modeling*

Résumé

Contributions aux ombres et jumeaux numériques dans l'industrie : proposition d'une stratégie de couplage entre modèles de simulation et d'apprentissage automatique appliquée aux scieries

Ces travaux de thèse s'inscrivent dans le projet ANR Lorraine-Intelligence Artificielle qui se veut un projet multi-disciplinaire promouvant la recherche à la fois sur l'intelligence artificielle elle-même et sur ses applications à d'autres domaines de recherche. A ce titre, cette thèse s'intéresse au développement et à l'utilisation de modèles d'apprentissage automatique comme modèles de substitution à des modèles de simulation. L'intérêt pour ce sujet de recherche est, en particulier, porté par l'engouement des milieux académiques et industriels pour le concept d'ombres et jumeaux numériques, vus comme une évolution des modèles de simulation pour une utilisation pérenne au cœur des systèmes et des processus.

La contribution principale de ces travaux de thèse est la proposition d'une stratégie de couplage entre un modèle de simulation et un modèle de substitution réalisant une même tâche de prédiction de manière répétée sur un flux de données. Le modèle de simulation est supposé avoir un haut niveau de fidélité mais être trop lent ou coûteux en calcul pour être utilisé seul pour réaliser l'intégralité des prédictions requises. Le modèle de substitution est un modèle d'apprentissage automatique qui approxime le modèle de simulation. L'objectif premier de la stratégie de couplage proposée est l'utilisation efficiente des ressources en calcul limitées par l'allocation intelligente de chaque prédiction à effectuer à un des deux modèles. Cette allocation est, en particulier, inspirée de l'apprentissage actif et basée sur l'évaluation de niveaux de confiance dans les prédictions du modèle d'apprentissage automatique. Des expériences numériques sont d'abord menées sur huit jeux de données de la littérature scientifique. Une application à l'industrie du sciage est ensuite développée.

Mots-clés: *Scierie*, *Apprentissage automatique*, *Apprentissage actif*, *Ombre numériques*, *Modèles de substitution*

Table des matières

Table des figures

Liste des tableaux

Acronymes

ACP	Analyse en Composantes Principales
AL	Active Learning
ANR	Agence Nationale de la Recherche
ANN	Artificial Neural Network
EPC	Erreurs Préquentielles de Couple
EPPNS	Erreurs Préquentielles des Prédictions des points de donnés Non Simulés
ESF	Ensemble of Shape Functions
EWMA	Exponential Weighted Moving Average
JN	Jumeau Numérique
ICP	Iterative Closest Point
kNN	k Nearest Neighbors
MLP	Multi-Layered Perceptron
MSE	Mean Squared Error
M2DP	Multiview 2D Projection
O&JN	Ombres et Jumeaux numériques
ON	Ombre Numérique
PdP	Paniers de Produits
RMSE	Root Mean Squared Error
R&D	Recherche et Développement

UCI University California Irvine

UML Unified Modelling Language

Chapitre 1

Introduction

Les travaux de thèse présentés dans ce mémoire ont été réalisés au sein du laboratoire multidisciplinaire CRAN [1]. Ils s'inscrivent dans la continuité des travaux du département ISET (Ingénierie des Systèmes Eco-technique) sur les systèmes industriels, et sur les jumeaux numériques. Le développement de cet axe de recherche est, en particulier, une volonté affichée de l'équipe S&O2I à laquelle j'ai été intégré.

De plus, cette thèse s'inscrit dans le cadre du projet ANR Lorraine Artificial Intelligence [2]. Ce projet a deux objectifs principaux. Le premier est de renforcer l'excellence des travaux menés par l'université de Lorraine en intelligence artificielle et dans les domaines d'application qui lui sont associés, comme la santé ou l'industrie. Ainsi, ce projet promeut une approche interdisciplinaire qui sous-tend de nombreuses actions de l'université de Lorraine. Le deuxième est d'ouvrir de nouveaux pans de recherche en intelligence artificielle. Au total, 24 thèses sont co-financées par ce projet.

Ainsi, les travaux présentés dans ce mémoire traitent de l'utilisation de l'apprentissage automatique pour la création et la coordination de modèles numériques intégrés au sein d'ombres et jumeaux numériques pour la réalisation de tâches de prédiction. Par ailleurs, une part importante des travaux présentés porte sur une application à l'industrie forêt-bois et plus particulièrement aux scieries.

1.1 Ombres et jumeaux numériques

Ombres et jumeaux numériques sont des termes parapluies pour lesquels de nombreuses définitions ont été proposées. Ainsi la définition même de ces concepts reste un sujet de recherche actif. Une définition très générale, proposée par exemple par Savolainen et Knudsen, 2022, présente les jumeaux numériques comme un ensemble de modèles numériques d'entités physiques capables d'exploiter des données collectées en temps réel pour l'optimisation et l'aide à la décision. Bien que les jumeaux numériques soient souvent associés aux modèles de simulation, les modèles qui

1. http://www.cran.univ-lorraine.fr/
2. https://anr.fr/Project-ANR-20-THIA-0010

leur sont associés peuvent être de nature diverse, que ce soit une modélisation explicite du système réel ou des algorithmes d'apprentissage automatique.

De nombreux verrous ont été soulevés par la littérature scientifique étudiant ces concepts. En particulier, Julien et Hamzaoui, 2023 évoque la prise en compte de la sobriété digitale par les ombres et jumeaux numériques et la coordination de plusieurs modèles ayant des coûts en calculs et des niveaux de fidélités différents. En effet, différents types de modèles présentent, en général, différents avantages et inconvénients. Les modèles de simulation, par exemple, basés sur des connaissances et représentations explicites du système réel, sont souvent relativement interprétables et peuvent être très fidèles à la réalité une fois calibrés. En particulier, ils peuvent être utilisés pour étudier des scénarios jusque-là jamais observés, dans la limite du domaine de validité qui leur est associé. Ces caractéristiques sont, cependant, souvent contrebalancées par un coût en calcul important. Les modèles d'apprentissage automatique, au contraire, peuvent souvent être exécutés très rapidement mais souffrent de leur image de boite noire et restent difficiles à interpréter. De même, la qualité des prédictions faîtes par ces modèles est très dépendante des données ayant été utilisées pour les entraîner. De fait, les modèles d'apprentissage automatique peuvent rencontrer des difficultés à généraliser à des données trop différentes de celles sur lesquelles ils ont été entraînés (Ritto et Rochinha, 2021). Par ailleurs, les temps de calculs importants associés aux modèles de simulation ont, en particulier, poussé de nombreux chercheurs à étudier la possibilité de créer des modèles simplifiés utilisables comme des approximations rapides à évaluer des modèles de simulation. Ces modèles, souvent issus des domaines de l'apprentissage statistique ou de l'apprentissage automatique, sont entraînés sur des résultats de simulation obtenus à l'aide de plans d'expériences ou par d'autres méthodes spécifiques. Il s'agît donc d'échanger la fidélité des modèles de simulation contre la rapidité de prédiction des modèles d'apprentissage.

1.2 Une application aux scieries

Les scieries sont des acteurs de la première transformation dans la filière forêt-bois. Elles transforment les grumes de bois en produits de sciage (poutres, planches, merrains...) et produits connexes (écorce, sciure, copeaux...). Ces produits connexes sont, pour leur grande majorité, valorisés dans des industries diverses, pour la production d'énergie, de panneaux de particules, de papiers, ou dans la chimie verte.

Les opérations de sciage ont la particularité d'être divergentes et en co-production. D'une seule et même grume de bois, une scierie obtiendra simultanément plusieurs produits de sciage et autres sous-produits. En particulier, elle obtiendra simultanément plusieurs produits de sciage ayant des dimensions et des qualités différentes. Dans la suite de cette thèse, nous appellerons panier de produits (PdP) l'ensemble des produits obtenus du sciage d'une grume.

Plusieurs facteurs rendent difficile la planification des opérations de sciage. D'une part, le fait que le processus soit en co-production implique que la production d'un type de produit générera d'autres produits annexes, possiblement de faible valeur, qui devront être stockés. Par ailleurs, l'hétérogénéité de la matière première, en termes de forme des grumes ou de défauts internes rend difficile l'anticipation du mix-produit obtenu après sciage d'un lot de grumes. Cette incertitude est exacerbée par les caractéristiques des lignes de sciage modernes qui ajustent automatiquement le plan de coupe grume par grume.

Deux grandes familles de solutions ont été proposées dans la littérature pour minimiser l'impact de cette incertitude sur la planification de la production (Zanjani, Nourelfath et Ait-Kadi, 2013b). La première est basée sur l'utilisation de méthodes de recherche opérationnelle permettant de prendre en compte cette incertitude dans le processus d'optimisation. La deuxième est basée sur l'utilisation d'outils de simulation, permettant d'estimer le PdP de grumes individuelles à partir d'informations sur leur forme ou leurs défauts internes. Par exemple, les simulateurs Optitek ou SAWSIM peuvent simuler le processus de sciage d'une large catégorie de scieries et estimer le panier de produits de grumes à partir de scans 3D de leur surface externe. Il est intéressant de noter que ces deux approches peuvent être combinées. Par exemple Zanjani, Nourelfath et Ait-Kadi, 2013b utilise un simulateur pour générer les paramètres d'un problème de programmation stochastique à partir d'un échantillon de grumes. Cependant, les simulateurs de sciage peuvent être sujets à une trop grande complexité temporelle pour être utilisable dans des problèmes opérationnels pouvant impliquer des milliers de grumes. C'est ce qui a poussé des chercheurs du FORAC, puis du CRAN, à travailler au développement de modèles de substitution pour ces simulateurs (Morin et al., 2015 ; Selma et al., 2018). Ces modèles de substitution, aussi appelés métamodèles, sont des modèles de substitution basés sur des algorithmes d'apprentissage automatique entraînés sur des résultats de simulation passés à prédire le PdP de nouvelles grumes. L'avantage principal de ces métamodèles est leur capacité à fournir une prédiction extrêmement rapidement. Ils présentent cependant un certain nombre d'inconvénients. En particulier, la construction de la base d'exemples de résultats de simulation reste coûteuse car elle requiert d'effectuer plusieurs centaines voire plusieurs milliers de simulations. Cela est rendu d'autant plus problématique par le fait que ces métamodèles ne sont valides que pour une seule configuration de scierie et liste de prix de produits (utilisées pour construire la base d'exemple de simulation). Ces métamodèles ont donc une utilisation limitée dans le temps, d'autant plus courte que les listes de prix sont mises à jour régulièrement. De plus, les PdP prédits par ces métamodèles restent des approximations des vrais résultats de simulation, dont l'exactitude varie d'un modèle de scierie à un autre et suivant la taille et la qualité de la base de données d'entraînement.

Dans le cas des scieries, une ombre numérique capable de prédire les PdP de grumes entrant en scieries, et d'utiliser ces prédictions pour proposer des plans optimisés de production au niveau opérationnel pourrait considérablement aider la planification de la production en scieries. Plusieurs problèmes se posent cependant quant au développement de tels systèmes. En particulier, les temps de simulation requis par les simulateurs de sciage rendent difficile leur adaptation au contexte des ombres numériques.

1.3 Originalité de la thèse

La recherche effectuée durant cette thèse s'intéresse, à ce titre, au développement d'un modèle d'ombre numérique basé sur le couplage d'un modèle de simulation et d'un modèle de substitution dans le cas ou le modèle de simulation est trop coûteux en calcul pour être utilisé seul. L'objectif est, plus particulièrement, de développer une stratégie pouvant tirer parti à la fois de la vitesse de prédiction des métamodèles et de la précision des simulateurs. Les verrous scientifiques principaux qui seront traités durant cette thèse seront l'adaptabilité des jumeaux numériques au changement, ainsi que le coût important de labélisation des données. Les contributions visées sont les suivantes :

— La proposition d'une stratégie permettant de coupler un modèle de simulation (ou plusieurs instances de ce modèle) et son métamodèle au sein d'une ombre numérique. Cette stratégie

s'appuie, en particulier, sur l'apprentissage actif, un champ de l'apprentissage automatique impliquant les modèles eux-mêmes dans la sélection des données qui servent à les entraîner.

— L'étude des propriétés théoriques de la stratégie proposée, et son application sur 8 jeux de données de référence UCI. Pour ces expériences, l'algorithme d'apprentissage automatique utilisé est une forêt aléatoire (Breiman, 2001), et le problème de prédiction est un problème de régression.

— L'intégration de méthodes de détection de dérive conceptuelle à la stratégie proposée, et la mise en évidence de l'effet du biais d'échantillonnage induit sur les performances de ces méthodes de détection.

— L'application de la stratégie proposée à un problème issue de l'industrie du sciage. Le problème de prédiction concerne la prédiction de PdP de grumes de bois à partir de scans 3D de leurs surfaces, et est traité comme un problème de régression. Le jeu de données utilisé a été fourni par le FORAC et provient de la filière forêt-bois canadienne.

Chapitre 2

Ombres numériques, jumeaux numériques et apprentissage automatique

2.1 Introduction

La littérature attribue plusieurs origines au concept de jumeau numérique (JN, *Digital Twin* en anglais) duquel dérive celui d'ombre numérique (ON, *Digital Shadow* en anglais). Certains auteurs, comme Boschert et Rosen, 2016, citent le projet Apollo de la Nasa au cours duquel deux véhicules identiques avaient été construits. L'objectif était de reproduire sur terre les conditions du véhicule envoyé dans l'espace pour émuler et résoudre des problèmes auquel il pouvait être confronté. D'autres auteurs, comme Singh et al., 2021 et Grieves, 2014 évoquent un cours de gestion du cycle de vie des produits donné à l'université du Michigan en 2003.

Quoi qu'il en soit, la première définition du terme jumeau numérique semble avoir été proposée en 2012 par la Nasa dans Shafto et al., 2012 comme suit : "A digital twin is an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. The digital twin is ultra-realistic and may consider one or more important and interdependent vehicle systems, including propulsion/energy storage, avionics, life support, vehicle structure, thermal management/TPS, etc. In addition to the backbone of high-fidelity physical models, the digital twin integrates sensor data from the vehicle's on-board integrated vehicle health management (IVHM) system, maintenance history, and all available historical/fleet data obtained using data mining and text mining. The systems on board the digital twin are also capable of mitigating damage or degradation by recommending changes in mission profile to increase both the life span and the probability of mission success." Cette définition est, cependant, très spécifique au cas d'application pour lequel il a été conçu par la Nasa. Ce concept a, ainsi, été étendu à de nombreux autres types d'homologues physiques pour des utilisations variées.

Depuis cette première définition, l'intérêt du monde académique comme du monde industriel

pour les jumeaux numériques n'a cessé de grandir. A titre d'illustration, la figure 2.1 présente le nombre de documents référencés dans la base scientifique scopus, publiés entre 2012 et 2022 qui contiennent le terme "digital twin" dans leur titre, résumé ou mots-clés. Comme illustré par cette figure, le nombre d'articles traitant de ce sujet publiés chaque année croit très rapidement depuis 2017.

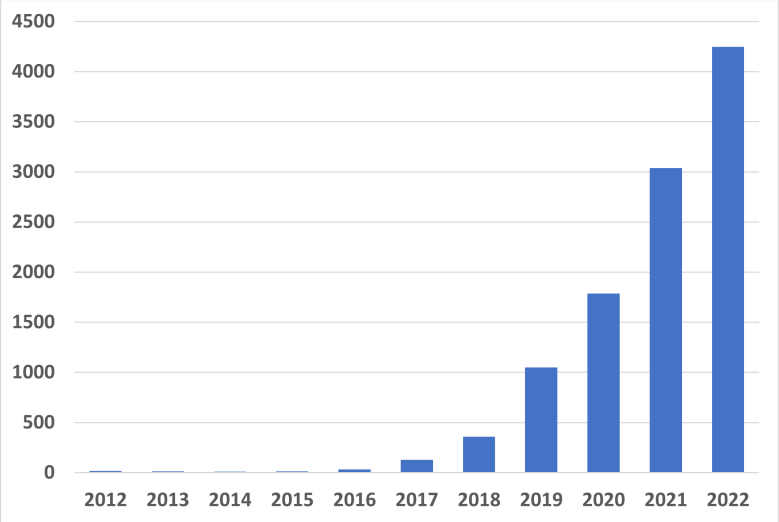

FIGURE 2.1 : Nombre de documents référencés par la base scientifique scopus entre 2012 et 2022 contenant le terme "digital twin" dans son titre, son résumé, ou ses mots-clés.

Les ombres et jumeaux numériques sont donc un domaine de recherche actif ayant trouvé des applications dans de nombreux secteurs, industriels ou non, comme illustré dans la figure 2.2. Ce diagramme circulaire présente les domaines des documents obtenus par la recherche précédente. On y retrouve des domaines aussi variés que l'ingénierie, l'énergie, les sciences sociales ou les sciences de l'environnement.

Pourtant, la définition exacte du JN fait toujours débat et de nombreuses définitions, souvent spécifiques à leur domaine d'étude, ont été proposées. SEMERARO et al., 2021, en particulier, proposent une revue systématique de la littérature recensant 30 définitions différentes publiées entre 2012 et 2019 dans la littérature scientifique. SAVOLAINEN et KNUDSEN, 2022 identifient deux origines aux écoles sous-jacentes à ces nombreuses définitions. La première est basée sur la capacité croissante des modèles de conception assistée par ordinateur à produire des représentations extrêmement fidèles de la structure et du comportement de systèmes physiques complexes. Les définitions s'appuyant sur cette école insistent donc sur la capacité du JN à décrire la géométrie, l'historique et le comportement du système réel. Par exemple GRIEVES et VICKERS, 2017 proposent la définition suivante "the Digital Twin is a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level. At its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin". Cette définition pousse à l'extrême l'idée du JN comme une représentation parfaitement fidèle du système physique. D'autres auteurs préfèrent voir le JN comme une évolution des systèmes de simulation ou d'émulation, élargissant leur utilisation du secteur de la R&D au secteur opérationnel. Les définitions s'appuyant sur cette seconde école insistent donc sur les

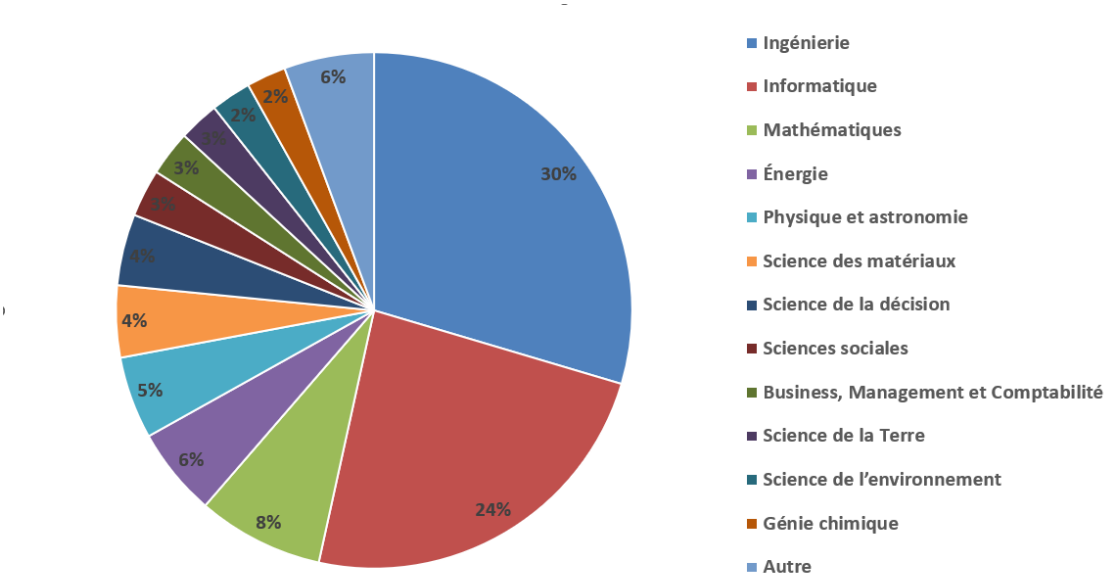

FIGURE 2.2 : Domaines des documents référencés par la base scientifique scopus entre 2012 et 2022 contenant le terme "digital twin" dans son titre, son résumé, ou ses mots-clés

capacités du JN à évaluer et comparer plusieurs scénarios, ou à prédire un état futur du système physique. Les données ainsi générées peuvent ensuite servir les objectifs spécifiques du jumeau, que ce soit l'optimisation d'un processus de production ou l'anticipation de pannes. Par exemple SEMERARO et al., 2021 proposent la définition suivante : "A set of adaptive models that emulate the behaviour of a physical system in a virtual system getting real time data to update itself along its life cycle. The digital twin replicates the physical system to predict failures and opportunities for changing, to prescribe real time actions for optimizing and/or mitigating unexpected events observing and evaluating the operating profile system". Cette thèse s'appuie principalement sur cette seconde vision du JN.

Le concept d'ombre numérique dérive de celui de jumeau numérique, et n'en est pas toujours différencié dans la littérature. De fait, comme remarqué par LIU et al., 2021 et KRITZINGER et al., 2018, de nombreuses études utilisant le terme "jumeau numérique" traitent en réalité d'ombres numériques ou de modèles numériques. La suite de ce chapitre est organisée comme suit. D'abord, la section 2.2 précise les concepts d'ombre et de jumeau numérique, leurs différences et points communs, et présente plusieurs typologies d'ombres et jumeaux numériques (O&JN). Dans un deuxième temps, la section 2.3 introduit plusieurs champs de recherches en apprentissage automatique et leurs liens avec celui des O&JN. L'objectif n'est pas de présenter un panel exhaustif de toutes les branches de l'apprentissage automatique ayant été appliquées dans le cadre des O&JN mais de présenter celles considérées dans la suite de cette dissertation.

2.2 Ombres et jumeaux numériques

2.2.1 Modèle numérique, ombre numérique et jumeau numérique

Plusieurs distinctions existent dans la littérature entre les concepts de *modèle numérique*, d'*ombre numérique* et de *jumeau numérique* à proprement parler. L'une des plus communes a été introduite par Kritzinger et al., 2018 et se base sur le niveau d'intégration entre l'entité numérique et son homologue physique, comme illustré figure 2.3.

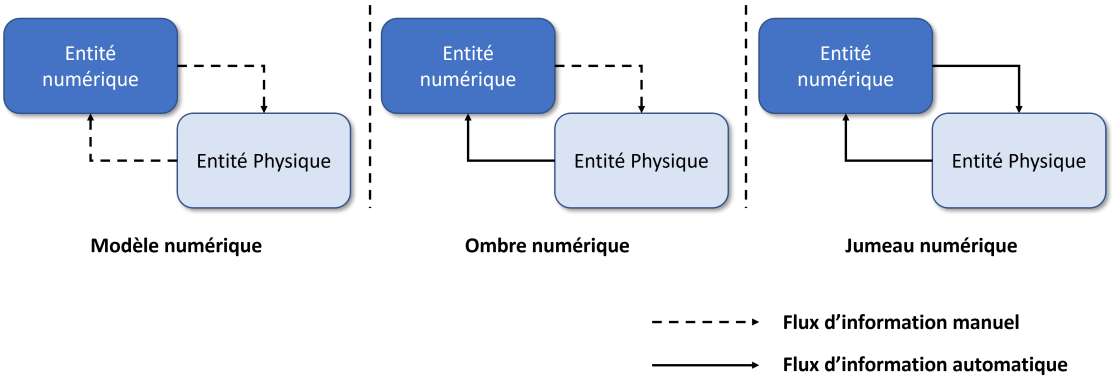

Figure 2.3 : Différence entre un modèle numérique, une ombre numérique et un jumeau numérique, d'après Kritzinger et al., 2018

Le niveau d'intégration fait ici référence à la nature automatique ou non des flux de données et d'informations entre l'entité physique et l'entité virtuelle.

— Dans le cas des modèles numériques, le flux d'informations n'est automatisé ni de l'entité physique vers l'entité virtuelle, ni de l'entité virtuelle vers l'entité physique. Tout transfert de données vers l'entité numérique pour l'analyse ou la mise à jour des modèles est déclenché par l'utilisateur. De fait, un modèle numérique ne peut s'adapter automatiquement à des changements de l'entité physique auquel il est associé. De même, le modèle numérique ne peut en aucun cas influer directement sur l'entité physique sans intervention humaine.

— Dans le cas des ombres numériques, le flux d'informations en provenance de l'entité physique vers l'entité numérique est automatique, mais le flux en provenance de l'entité numérique vers l'entité physique reste manuel. Cela signifie que l'ombre numérique est en mesure d'analyser en continu des données récoltées en temps réel ou quasi réel depuis le jumeau physique. Ces données peuvent permettre à l'ombre numérique de s'adapter si nécessaire à des changements ou perturbations observés au niveau de l'entité physique. En revanche, l'ombre numérique ne peut en aucun cas commander automatiquement des actions ou des changements sur l'entité physique. A ce titre, l'ombre numérique est avant tout un outil d'aide à la décision. Par exemple, Sapel et al., 2022 étudient une ombre numérique pour l'ordonnancement et le contrôle d'opérations de moulage par injection. L'ombre numérique proposée est reliée à une base de données contenant les historiques complets de production et les caractéristiques de plusieurs machines. Lorsque la production d'un type de pièces doit être planifiée, l'ombre est interrogée pour trouver la machine minimisant la production de pièces défectueuses. De même, Ladj et al., 2021 proposent une ombre numérique de

machines outils pour l'aide à la décision dans l'industrie manufacturière. L'ombre récolte en continu des données relatives aux actions de l'entité physique et à l'environnement. Ces données sont utilisées, notamment, pour la classification de l'état de la machine et la détection de défauts. Cependant, l'ombre numérique ainsi conçue ne peut influer sur la machine physique et n'est donc, ici, qu'une étape intermédiaire vers la conception d'un jumeau.

— Dans le cas du jumeau numérique, les flux de données sont automatiques dans les deux sens. Les changements aux niveaux de l'entité physique pourront donc être automatiquement détectés et reflétés par le jumeau numérique, et vice versa. Par exemple, Bottani et al., 2017 développent un jumeau numérique d'un véhicule guidé autonome. Celui-ci reçoit des informations de l'environnement du véhicule, sous la forme de tâches à réaliser et d'états de plusieurs machines. Le jumeau numérique est alors en mesure de sélectionner à quelle machine le véhicule doit livrer quelle pièce, et influer sur le parcours du véhicule en conséquence.

On peut remarquer que cette distinction entre modèles, ombres et jumeaux numériques ne fait en aucun cas intervenir le niveau de fidélité des représentations et modèles sous-jacents. Développer, maintenir et utiliser des modèles ayant un haut niveau de fidélité est, en effet, coûteux en ressource sans être nécessairement créateur de valeur (Julien et Hamzaoui, 2023). Les ombres et jumeaux numériques peuvent être caractérisées suivant plusieurs dimensions.

2.2.2 Caractéristiques des ombres et jumeaux numériques :

La littérature traitant des ombres et jumeaux numériques est très vaste et diversifiée. De nombreux types d'ombres et jumeaux ont, en particulier, été proposés par différents auteurs présentant des caractéristiques différentes suivant leurs contextes et utilisations. Beaucoup de ces distinctions sont indépendantes de la classification de l'entité numérique entre ombre et jumeau. Il est cependant nécessaire de les présenter pour positionner les travaux réalisés durant cette thèse. Les O&JN peuvent, en particulier, être catégorisés suivant leur intégration dans le cycle de vie de l'homologue physique, la nature de cet homologue physique, mais aussi par leur niveau d'abstraction et par la nature des modèles qu'ils contiennent.

Cycle de vie

Plusieurs définitions estiment que les ombres et jumeaux numériques doivent suivre leur homologue physique tout au long de son cycle de vie, depuis la phase de conception jusqu'à la fin de vie (Tuegel, 2012 ; Ríos et al., 2015 ; Schroeder et al., 2016 ; Haag et Anderl, 2018). A ce titre, Grieves et Vickers, 2017 distinguent deux types d'ombres (ou jumeaux) numériques suivant la phase du cycle de vie dans lequel se trouve l'homologue physique :

— L'**ombre ou jumeau numérique prototype** (initialement simplement nommée par les auteurs *Digital Twin Prototype*, ou instance de jumeau numérique) est créé durant la phase de conception de l'homologue physique. Il contient, entre autres, la liste du matériel et les instructions nécessaires à la conception de l'homologue physique, ainsi que les modèles de conception assistée par ordinateur. Il peut être utilisé pour tester divers scénarios d'utilisation et optimiser le design de l'homologue physique. Malgré le nom attribué par

Grieves et Vickers, 2017 à ce type d'entité numérique, la classification de Kritzinger et al., 2018 ne permet de le classifier que comme un modèle numérique.

— **L'instance d'ombre ou de jumeau numérique** (*Digital Twin Instance* en anglais) est utilisée après la création de l'homologue physique. Au contraire du jumeau numérique prototype, il est lié à l'entité physique qu'il suivra jusqu'à sa destruction. A ce titre, il est indispensable que les modèles inclus dans l'ombre numérique soient capables de s'adapter aux changements de son homologue physique.

Suivant la nature de l'homologue physique, son cycle de vie sera différent. La figure 2.4, par exemple, décrit les cycles de vie d'un produit et d'un processus de production. Bien que différentes, ces entités interagiront durant la phase de production. De même, les jumeaux numériques du produit et du système de production pourront être amenés à interagir durant la phase de production, par exemple en utilisant les données spécifiques à un produit pour en optimiser l'assemblage (Boschert et Rosen, 2016).

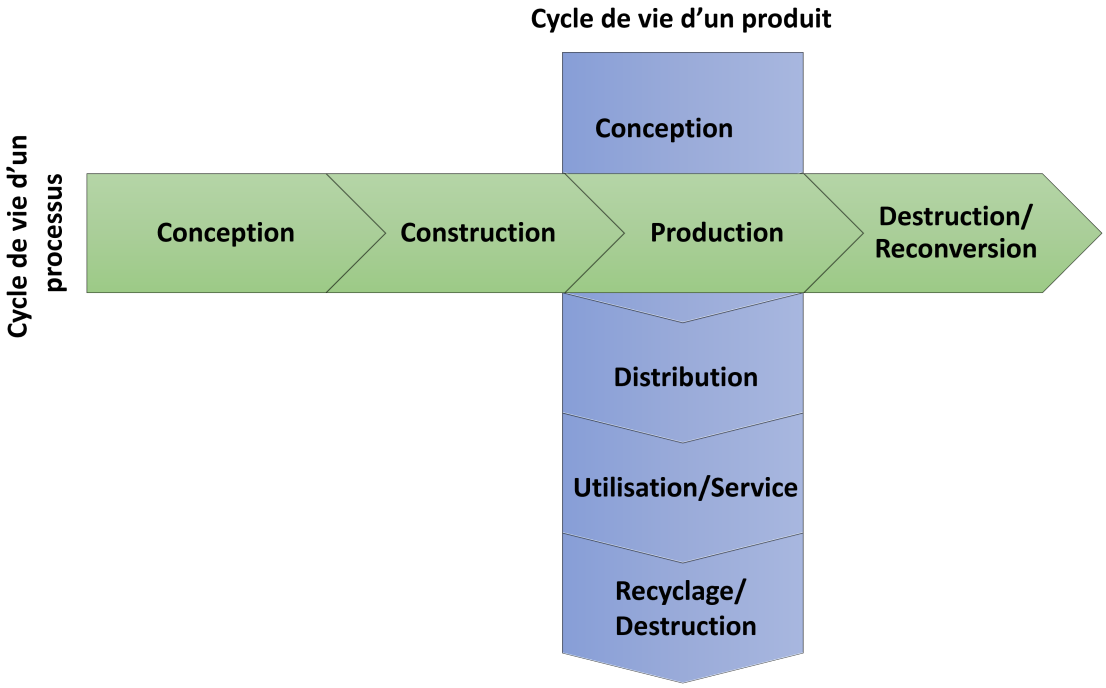

Figure 2.4 : Cycle de vie d'un processus de production et d'un produit

Nature de l'homologue physique

Des ombres et jumeaux numériques ont été associés à des entités physiques très diverses. Dans l'industrie en particulier, des ombres et jumeaux numériques ont été proposés pour des homologues physiques à des échelles très différentes allant du simple composant à l'usine entière. Par exemple, Wang et al., 2019 proposent un jumeau numérique d'un rotor pour la détection de pannes et la maintenance prédictive. Zhou et al., 2020 proposent un jumeau numérique pour une cellule

de production composée de plusieurs machines et zones de stockage. Azangoo et al., 2021 introduisent un jumeau numérique d'une usine de traitement d'eau.

En particulier, Ladj et al., 2021 identifient quatre types d'O&JN. Les trois premières sont les *O&JN de produits*, les *O&JN d'actifs* et les *O&JN de processus*. A ces trois catégories s'ajoutent les *O&JN d'humains*.

Les O&JN de produits peuvent être utilisés pour simuler et contrôler l'état et le comportement du produit durant les phases de conception et de production, puis durant la phase de service. Ils permettent de conserver les informations sur la géométrie et le processus de production individuel du produit à travers plusieurs étapes de la production et plus généralement plusieurs étapes du cycle de vie du produit.

Les O&JN d'actifs sont conceptuellement similaires aux O&JN de produits. Ils contiennent les données et modèles relatifs à une unité de production d'un atelier, par exemple un outil ou une machine. Ils peuvent, suivant les cas d'usage, contenir des informations sur la structure, l'état et l'environnement de l'homologue physique. De tels O&JN peuvent avoir des applications très diverses, que ce soit pour l'optimisation d'un réglage, la détection de défauts ou la prédiction de consommation en ressources. Plus généralement, il est envisageable d'implémenter des O&JN attachés à d'autres types de ressources de production.

Les O&JN de processus correspondent à des O&JN d'ateliers ou d'usines, comprenant un ou plusieurs actifs et à travers lesquels transitent matières premières, composants, et produits en cours de production. Ils peuvent être utilisés pour optimiser le processus et évaluer des solutions en cas de problèmes. Par exemple, Zhang et al., 2017 développent un JN pour une ligne de production de verre creux composée de plusieurs machines, véhicules et espaces de stockages. Le JN est d'abord créé durant la phase de conception de la ligne de production pour en optimiser la configuration, puis adapté pour le contrôle du processus.

Enfin, certains auteurs considèrent également des O&JN d'opérateurs humains. Cardin et Trentesaux, 2022, en particulier, décrivent un tel O&JN comme une représentation de l'état physique et mental de l'opérateur, ainsi que de son comportement.

Catégorisation des O&JN par niveaux d'abstraction

Une catégorisation des jumeaux numériques en trois niveaux hiérarchiques a été proposée par Tao et al., 2019 en lien avec la taille de l'homologue physique. Elle est soutenue par l'idée, défendue par Bril El Haouzi, 2017 ; Savolainen et Knudsen, 2022 que les ombres et jumeaux numériques devraient, autant que possible, être modulaires pour faciliter leur implémentation et leur adaptation à des changements au niveau de l'homologue physique. Ces trois catégories sont :

— *Le niveau élèmentaire* (*unit level* en anglais) fait référence à la plus petite unité physique pour laquelle une ombre ou jumeau numérique, même partiel est implémenté. Cette unité peut, par exemple, être un composant ou une machine. Les ombres et jumeaux numériques au niveau unitaire se basent sur la géométrie, les fonctionnalités et le comportement de l'homologue physique.

— *Le niveau système* (*system level* en anglais) considère les amalgames d'éléments unitaires, comme un atelier composé de plusieurs machines, zones de stockages, et flux. Une ombre

ou jumeau numérique de système est alors constituée de plusieurs ombres et jumeaux numériques d'unités et de leurs interactions. Le niveau système est adapté aux O&JN de processus qui font le lien entre plusieurs O&JN de ressources et de produits.

— *Le niveau système de systèmes* (*system of systems* level en anglais) est un amalgame de systèmes. Une O&JN d'un système de systèmes peut être utilisée pour favoriser la coopération entre plusieurs départements d'une entreprise ou plusieurs entreprises. Par exemple, Reitz, Schluse et Rossmann, 2019 étudient le développement d'un réseau de jumeaux numériques dans la chaîne d'approvisionnement de la filière forêt-bois, facilitant le partage d'informations sécurisées entre différents acteurs et l'optimisation des opérations.

Modèles basés sur les connaissances ou les données

Comme souligné par Semeraro et al., 2022, les modèles inclus dans une ombre numérique peuvent être séparés en deux catégories : ceux basés sur les connaissances (*knowledge-based* ou *physic-based* en anglais) et ceux basés sur les données (*data-based* en anglais).

La première catégorie rassemble les modèles fondés sur les connaissances explicites du développeur en rapport avec l'entité physique associée à l'ombre numérique. Ces modèles peuvent, par exemple, être des modèles de simulation basés sur des lois physiques, la théorie des files d'attente, ou d'autres modèles à évènement discrets. Ces modèles ne requièrent que peu de données pour être calibrés.

Les modèles orientés données, au contraire, nécessitent moins de connaissances sur l'entité physique mais nécessite une masse de données importante pour inférer les informations qui lui sont utiles. C'est par exemple le cas des modèles d'apprentissage supervisé qui s'entraînent sur des exemples passés à réaliser une tache spécifique. Plusieurs avantages et inconvénients sont couramment associés aux modèles basés connaissances ou données (Savadjiev et al., 2020 ; Ritto et Rochinha, 2021). En particulier, les modèles basés données, tels que les modèles d'apprentissage automatique, ont des temps d'exécution courts. Ils peuvent cependant souffrir de difficultés à généraliser à des données et à des scénarios trop différents de ceux ayant servi à leur entraînement. De plus leurs structures et paramètres sont difficilement interprétables. De même, les décisions prises par ces modèles sont souvent difficiles à expliquer. A ce titre, ces modèles sont souvent considérés comme des boîtes noires.

Au contraire, les équations et paramètres des modèles basés sur les connaissances ont, par définition, une interprétation physique. De plus, une fois calibrés et validés, ils ont de bonnes capacités d'extrapolation et peuvent être utilisés pour évaluer différents scénarios. Cependant, s'il est possible de définir des modèles basés sur les connaissances très fidèles aux systèmes réels, ces modèles peuvent être en pratique extrêmement coûteux en calculs. Par exemple, Spinti, Smith et Smith, 2022 évoquent le développement d'un modèle de simulation multi-physiques d'une chaudière industrielle, qui nécessite entre 5 et 7 jours de calculs à 1164 processeurs pour atteindre un l'état stationnaire requis pour l'exploitation des résultats.

2.2.3 Combinaison de modèles basés sur les connaissances et données

Les avantages et inconvénients des modèles basés sur les connaissances ou sur les données sont, ainsi, complémentaires. Il est donc naturel que de nombreuses études proposent des approches

pour combiner, coordonner ou coupler ces approches, que ce soit dans le cadre des O&JN ou d'autres.

Une première approche est l'intégration de modèles basés sur des algorithmes d'apprentissage automatique à l'intérieur d'un modèle plus large intégrant la connaissance du système physique. une telle approche peut être utilisée pour réduire le coût en calcul du modèle basé connaissance, ou si certains composants du système physique ne peuvent pas être modélisés efficacement. Par exemple, Azangoo et al., 2021 développent un jumeau numérique dit hybride d'une usine de traitement d'eau en remplaçant le modèle d'un élément (une chaudière) par un réseau de neurones récurrent. Les modèles des autres composants de l'usine et de leurs interactions restent basés sur la connaissance physique du processus. Dans un contexte différent, Thomas, Choffel et Thomas, 2008 proposent un modèle de simulation d'une scierie dont tous les composants à part les goulots d'étranglement sont remplacés par des réseaux de neurones.

De même, les résultats d'un modèle de simulation peuvent servir de données d'entrée à un modèle d'apprentissage automatique, et vice versa. Par exemple, Butt et Mohaghegh, 2022 proposent un jumeau numérique pour l'optimisation des réglages d'un processus d'impression 3D. Dans un premier temps plusieurs modèles intégrant la connaissance du fonctionnement de l'imprimante et du comportement physique des thermoplastiques sont utilisés pour déterminer la distribution de température à l'intérieur de la tête d'impression. Dans un second temps, cette distribution de température est utilisée par un modèle d'apprentissage automatique pour prédire la qualité des pièces produites. Dans un contexte différent, Shahhosseini et al., 2021 étudie l'utilisation de plusieurs modèles d'apprentissage automatique, dont des modèles de forêt aléatoires, pour améliorer la prédiction de rendements agricoles. Plus précisément, un modèle de simulation émet une première prédiction du rendement à partir de données d'entrée X. Cette prédiction et ces données d'entrée servent d'entrée à un modèle de forêt aléatoire qui émet une prédiction finale pour le rendement. Dans le cas étudié, les auteurs observent une réduction de la racine de l'erreur quadratique moyenne des prédictions entre 7% et 20%.

Enfin, une troisième approche est l'utilisation de modèles basés sur les connaissances pour entraîner des modèles d'apprentissage automatique. Lorsque ces modèles d'apprentissage sont entraînés à réaliser la même tâche de prédiction que le modèle basé connaissance, on parle de modèle de substitution ou de métamodèle. Plus généralement, Wang et al., 2022 proposent une revue de littérature sur différentes manières de combiner des modèles basés sur des connaissances ou des données pour des applications industrielles.

2.2.4 Verrous aux développement d'O&JN

De nombreux verrous scientifiques restent à lever pour accélérer le développement et l'utilisation des O&JN, dans le secteur industriel ou ailleurs, ont été identifiés par la littérature scientifique.

Par exemple, Semeraro et al., 2021 ; Singh et al., 2021 ; Tao et al., 2019 ; Julien et Martin, 2020 insistent sur l'importance de considérer les éventuelles interactions entre l'utilisateur humain et les O&JN, que ce soient par le développement d'interfaces ergonomiques ou en concevant le système non pas comme un jumeau numérique mais comme un triplet physique-numérique-humain. En particulier, Julien et Martin, 2020 insiste sur l'importance d'associer les utilisateurs au processus de développement du jumeau pour recueillir leur avis, cibler au mieux leurs besoins et faciliter l'acceptation de jumeau par la suite.

De même, Semeraro et al., 2021 ; Savolainen et Knudsen, 2022 ; Julien et Martin, 2020 mentionnent tous très justement la complexité de développer et maintenir les O&JN, en particulier au niveau système et système de systèmes. Considérant l'importance des ressources requises, ils recommandent en particulier de concevoir les O&JN suivant une architecture aussi modulaire que possible. Ainsi, une O&JN ne devrait pas être conçu comme un unique modèle très complexe mais comme un ensemble de modèles en interaction, qu'ils soient relatifs à des composants différents de l'homologue physique ou servent des objectifs différents. Ainsi, une structure modulaire permet un déploiement rapide des premiers éléments de l'O&JN développés, complétés aux fur et à mesure que de nouvelles fonctionnalités et modèles sont créées suivant les besoins établis. Une telle structure modulaire permet également de varier la granularité et la fidélité des modèles utilisés suivant les besoins (Savolainen et Knudsen, 2022). Il est également important de standardiser les modèles et structures de données utilisées, d'une part pour que l'O&JN ne soit pas lié à un fournisseur spécifique, et d'autre part pour faciliter les interactions entre modèles.

Ainsi, un troisième verrou est le dimensionnement des modèles inclus dans l'O&JN en fonction des objectifs de celui-ci. Contrairement à l'idée reçue selon laquelle les modèles inclus dans les O&JN doivent posséder un haut niveau de fidélité et/ou être adaptés au traitement de données massives, Julien et Martin, 2020 ; Julien et Hamzaoui, 2023 défendent le concept de lean data et de sobriété numérique, y compris dans le cadre des O&JN. Ces concepts impliquent, en particulier, une utilisation des modèles les plus complexes réduite aux seuls cas où ils sont absolument nécessaires et créateurs de valeur ajoutée. La coordination de plusieurs modèles ayant des niveaux de fidélités différents devient, alors, un défi spécifique. Julien et Hamzaoui, 2023 proposent, par exemple, l'utilisation de plusieurs modèles ayant des complexités et fréquences de synchronisation différentes, et de passer d'un modèle à l'autre suivant une mesure de cohérence entre valeurs prédites et observées. La solution qu'ils proposent implique, cependant, qu'entre deux changements, un même modèle est utilisé pour traiter l'intégralité des données transmises a l'O&JN, quelles qu'elles soient. Par ailleurs, peu d'articles traitent de tels mécanismes de coordination entre plusieurs modèles ayant des niveaux de fidélité différents.

Les verrous spécifiques traités durant cette thèse sont en lien avec ce troisième verrou. Parmi les approches utilisées, dans un contexte plus général que les O&JN, pour réduire le coût d'exécution de modèles numériques se trouvent en effet l'utilisation de modèles d'apprentissage automatiques comme modèles de substitution. Ces modèles de substitutions ont des niveaux de fidélité, coûts en calcul et besoins en données différents des modèles initiaux. Cependant, seule la création de ces modèles de substitution est généralement décrite et leur utilisation, en conjonction ou non avec le modèle initial est peu explorée. Nous traiterons donc ici de la combinaison de ces modèles de simulation et de leurs modèles de substitution, dans le cadre des O&JN pour proposer une méthode de couplage permettant de bénéficier de leurs avantages respectifs.

2.3 Apprentissage automatique pour les ombres numériques

2.3.1 Modèles de substitution/Métamodèles

Parmi les approches hybrides entre modèles basés connaissances ou données se trouve l'utilisation de modèles d'apprentissage automatique comme modèles substituts de modèles de simulation. On parle également de métamodèles. En anglais, on parle de *surrogate model* et de *metamodel*. Ce type de méthode peut être utilisé dans le cas où un modèle de simulation existe, mais est

trop coûteux en calcul pour être utilisé en pratique. Ce problème peut être amplifié lorsque plusieurs milliers de simulations sont requises, par exemple pour optimiser un processus utilisant une stratégie de simulation-optimisation (Xu et al., 2015). L'objectif est d'entraîner un modèle d'apprentissage automatique à effectuer la même tâche que le modèle de simulation, qui ne sert alors qu'à générer des exemples d'apprentissage. Le modèle d'apprentissage automatique est donc ici conçu comme une approximation du modèle de simulation, plus rapide à exécuter. Cependant, il s'agit bien uniquement d'une *approximation* du modèle initial, par conséquent moins précise. De plus, ce modèle d'apprentissage automatique pourra également souffrir des inconvénients courants pour ce type de modèles, à savoir, leur difficulté à généraliser à des données trop différentes de leurs bases d'entraînement. Cela est d'autant plus vrai que, par définition, la construction de la base servant à l'entraînement de ces modèles fait intervenir un modèle de simulation coûteux en calcul ce qui limite le nombre d'exemples d'apprentissage du modèle. De plus, le modèle d'apprentissage ne peut apprendre que des scénarios de simulation utilisés pour construire sa base de donnée d'entraînement. L'addition de nouveaux scénarios, le changement de paramètres de simulation non pris en compte précédemment ou des changements structurels du modèle de simulation (comme l'ajout ou le retrait de serveurs dans un réseau de files d'attente) rendraient nécessaire la mise à jour du modèle de substitution.

Principe

Les modèles substituts peuvent être séparés en deux grandes catégories : les métamodèles de surface de réponse (*response surface surrogates* en anglais), et les modèles à fidélité réduite (*lower-fidelity surrogates* en anglais) (Razavi, Tolson et Burn, 2012).

Les métamodèles de surface de réponse sont des fonctions cherchant à relier directement les paramètres d'entrée d'un modèle avec les sorties leurs étant associées. Ce sont des modèles basés données, ajustés sur une base d'exemples de résultats de simulation. Ils n'intègrent aucune connaissance du système spécifiquement modélisé. Cette base d'exemples peut être obtenue en réalisant des simulations sur des points échantillonnés au hasard dans le domaine souhaité de l'espace des entrées, en utilisant des plans d'expériences, ou par apprentissage actif. L'apprentissage actif, en particulier, est le domaine de l'apprentissage automatique qui étudie la sélection de points de données dans un ensemble de données non labélisées pour l'entraînement de modèles d'apprentissage automatique. Par exemple, Dasari, Cheddad et Andersson, 2019 développent des modèles substituts pour un simulateur permettant l'évaluation de plusieurs indicateurs de performance de turbines d'avions. Ces modèles substituts sont des modèles de type forêt aléatoire, entraînés sur une base d'exemples construite à l'aide d'un plan d'expériences en carré latin.

Les modèles à fidélité réduite sont des modèles basés sur les connaissances du système étudié, mais simplifiés par rapport au modèle initial. Par exemple, Thomas et Thomas, 2011 proposent un modèle de substitution pour un modèle de simulation de scierie, basé uniquement sur des réseaux de neurones artificiel (*Artificial Neural Networks* en anglais, ANN) et la simulation des goulots d'étranglements utilisant le logiciel Arena. Les Réseaux de neurones sont ici utilisés pour prédire les temps d'arrivée des produits circulant entre les goulots. Individuellement, ils sont des modèles d'apprentissage automatique utilisés pour modéliser la surface de réponse d'une variable interne à la simulation. Le modèle substitut complet fait cependant intervenir la connaissance du processus physique se déroulant dans la scierie.

Cas des ombres et jumeaux numériques

L'utilisation de modèles substituts est appropriée dans le cas des O&JN qui doivent traiter rapidement de grandes masses de données collectées en continu depuis leur homologue physique mais contiennent des modèles longs à exécuter. Ces méthodes sont, par ailleurs, considérées comme prometteuses par Savolainen et Knudsen, 2022 pour le développement d'O&JN opérationnels.

Ce type de méthode a été utilisé pour le développement de jumeaux numériques dans des domaines très variés. Un exemple intéressant est donné par Spinti, Smith et Smith, 2022. Il décrit la mise en place d'un jumeau numérique d'une chaudière industrielle à biomasse. Ce jumeau numérique est basé sur une simulation multi-physiques complexe de la chaudière, supposée avoir un haut niveau de fidélité. Cependant, ce modèle requiert un temps et une puissance de calcul considérable pour atteindre un état stationnaire nécessaire à l'extraction de données. Pour une utilisation opérationnelle, ce modèle initial a donc été remplacé par une collection de modèles substituts permettant de prédire les différentes grandeurs d'intérêt générées par la simulation. Plus particulièrement, ces modèles substituts sont des processus gaussiens, sélectionnés pour leur interprétation probabiliste, permettant une estimation de l'incertitude des prédictions. Les auteurs ne considèrent pas, cependant, la possibilité de faire évoluer ces modèles dans le temps. De même, Hürkamp et al., 2021 proposent une application pour le développement de jumeaux numériques de chaînes de processus dans l'industrie manufacturière, pour la production de pièces en thermoplastiques composites. Les simulations par éléments finis étant, en pratique, trop lentes pour pouvoir optimiser les paramètres des différents processus de production, les auteurs proposent de les remplacer par des modèles substituts. Ces modèles sont, dans ce cas, des forêts aléatoires entraînées à prédire un indicateur de la qualité de la pièce produite en fonction des paramètres des processus de fabrication. Les paramètres des simulations utilisés pour construire la base d'entraînement sont sélectionnés à l'aide d'un plan d'expériences. L'utilisation de ces modèles substituts dans un environnement opérationnel sujet à des perturbations pouvant influer sur la qualité des pièces produites est ensuite validée par une simulation à évènements discrets. Une application en science de l'environnement est proposée par Pylianidis et al., 2022. Leur motivation première, plus que l'implémentation de modèles rapides à exécuter, est de pouvoir implémenter ces modèles d'apprentissage automatique même en l'absence de données collectées sur le terrain. En particulier, le méta-modèle entraîné n'utilise pas certaines données nécessaires au modèle de simulation. Ces données sont en effet jugées trop difficiles à obtenir avec précision au niveau opérationnel. C'est par exemple le cas des prédictions météorologiques.

2.3.2 Apprentissage actif

L'apprentissage actif est le champ de l'apprentissage automatique qui étudie la sélection des points de données sur lesquels entraîner un algorithme d'apprentissage supervisé, sans connaissance a priori du label spécifique de chaque point. Les méthodes existantes dans la littérature font souvent intervenir le modèle en cours d'entraînement dans ce processus de sélection, d'où le nom d'apprentissage *actif*. Ce type de méthodes est utilisé lorsque les données non labélisées sont faciles à obtenir en grande quantité, mais que le processus de labélisation lui-même est coûteux, par exemple quand il mobilise une équipe d'experts ou une simulation computationnellement lente. L'objectif est donc de sélectionner parmi la masse de données non labélisées celles qui semblent

a priori les plus intéressantes pour l'entraînement du modèle d'apprentissage automatique. On appelle *oracle* l'entité responsable de la labélisation des données sélectionnées. Cette entité peut être un expert humain, un système réel ou un modèle numérique.

D'après la littérature plusieurs éléments doivent être déterminés préalablement à la mise en place d'une méthode d'apprentissage actif pour un problème industriel. Comme illustré figure 2.5, ces éléments sont le scénario d'échantillonnage, le problème d'apprentissage et le budget.

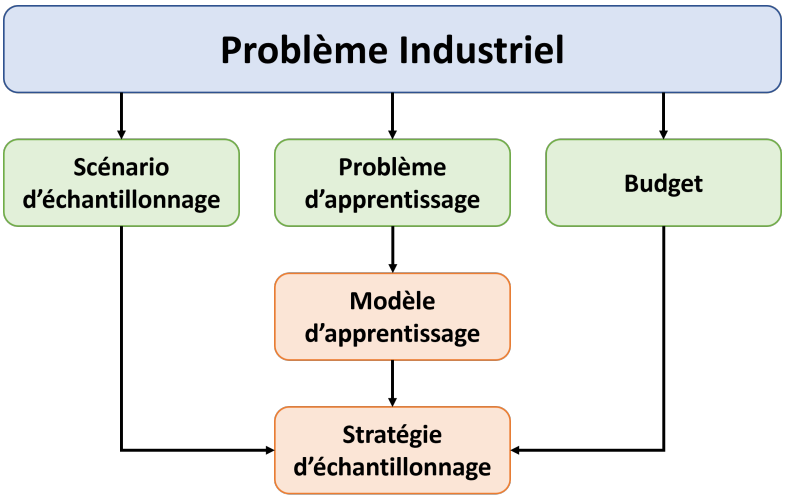

FIGURE 2.5 : Choix d'une stratégie de sélection de données à labéliser dans le cadre de l'apprentissage actif, CHABANET, EL-HAOUZI et THOMAS, 2021

Le scénario d'échantillonnage

Le scénario décrit les modalités d'accès aux données non labélisées lors de leur sélection. Trois types de scénarios existent dans la littérature :

— *Le scénario d'échantillonnage de base de données* (*pool-based active learning* en anglais) est considéré lorsque les données non labélisées sont rassemblées au sein d'une base de données. Cette base est accessible dans son entièreté dès le début des expériences. Cette base est fixe et ne sera à aucun moment complétée par de nouvelles données. Cependant, la sélection des données à labéliser est faite en utilisant la connaissance complète des points de données disponibles et de leur distribution spatiale.

Les processus de sélection associés à ce scénario sont souvent itératifs, comme illustré figure 2.6. Un critère mesurant l'intérêt de labéliser chaque point de données individuellement est calculé, et un ou plusieurs points optimisant ce critère sont sélectionnés. Ces points sont labélisés et utilisés pour entraîner à partir de zéro ou mettre à jour un modèle d'apprentissage automatique. Ce modèle peut ensuite être utilisé pour recalculer les critères mesurant l'intérêt de labéliser chaque point. De nouveaux points à labéliser sont ainsi sélectionnés et ainsi de suite jusqu'à ce qu'un critère de fin soit atteint.

— *Le scénario d'échantillonnage de flux de données* (*stream-based active learning* en anglais). Dans ce scénario, les données ne sont pas toutes accessibles simultanément dès le début du

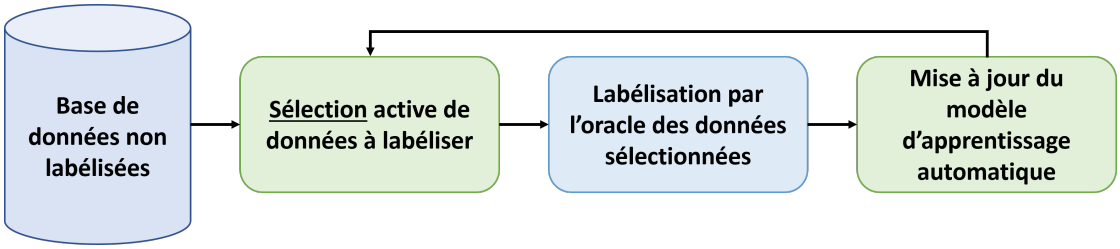

FIGURE 2.6 : Scénario d'échantillonnage de base de données

processus de sélection. De nouveaux points de données sont, au contraire, continuellement générés par un flux. Ce flux étant considéré infini, la décision de labéliser ou non un point de données doit être rapide. Certains auteurs, comme par exemple ŽLIOBAITĖ et al., 2013, étudient le cas extrême ou la décision de labéliser ou non un point de données doit être prise immédiatement, avant que le point suivant n'arrive. D'autres auteurs, comme IENCO et al., 2013, séparent le flux de données en blocs consécutifs et considèrent l'intégralité des données d'un bloc durant le processus de sélection. Un schéma présentant ce scénario est présenté figure 2.7. Comme pour le scénario d'échantillonnage de base de données, le modèle prédictif est régulièrement mis à jour à l'aide des données nouvellement labélisées par l'*oracle*. Suivant les cas, la prédiction du label par le modèle d'apprentissage automatique peut être réalisée avant ou après que la décision de labéliser le point de données par l'*oracle* ait été prise.

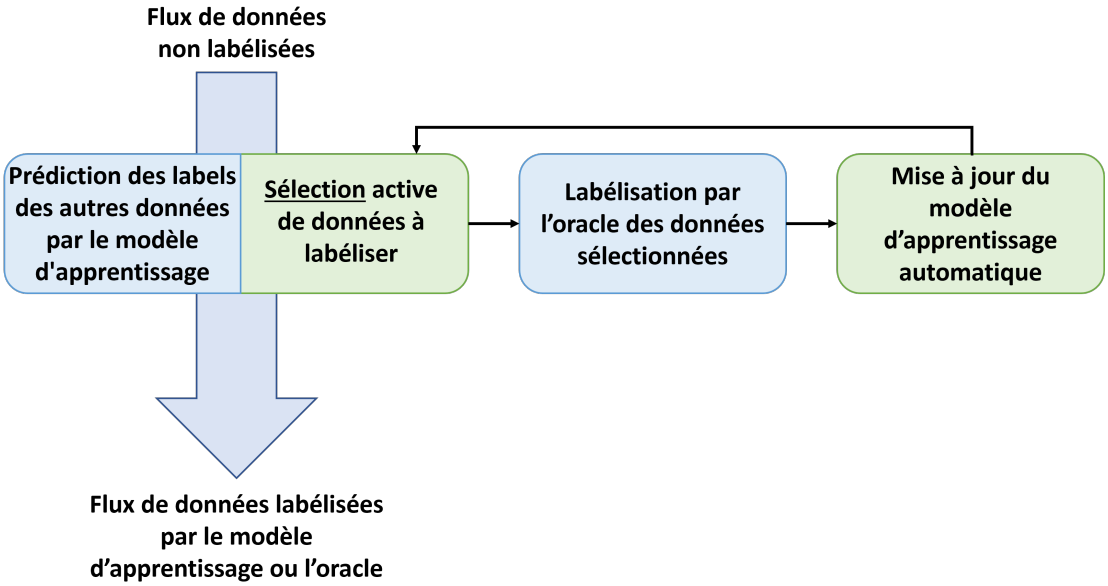

FIGURE 2.7 : Scénario d'échantillonnage de flux de données

— *Le scénario de génération de données* (*membership query synthesis active learning* en anglais) est le moins commun des trois scénarios d'apprentissage actif. Comme illustré figure 2.8, ce scénario ne fait intervenir aucune base ou flux de données. Les données servant à l'entraînement du modèle prédictif sont, au contraire, générées artificiellement avant d'être

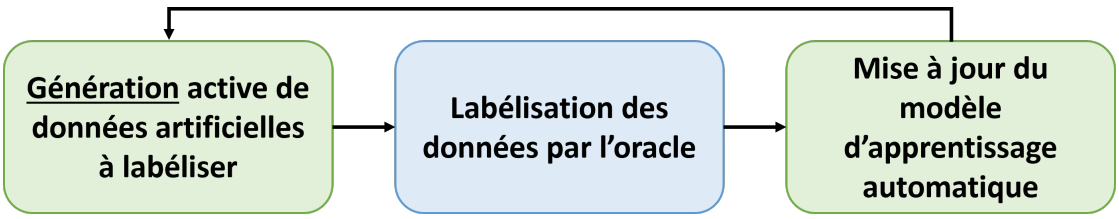

FIGURE 2.8 : Scénario de génération d'échantillons

labélisées par l'*oracle*. Ce scénario a été introduit à la fin des années 80 par ANGLUIN, 1988 qui considérait le problème d'identifier un sous-ensemble donné dans un ensemble plus grand. Ce type de stratégie peu permettre de générer rapidement de nouveaux points de données à labéliser mais souffre de problème d'interprétabilité des données synthétiques, en particulier lorsque l'*oracle* est humain (SETTLES, 2009). Pour remédier à ce problème, plusieurs auteurs comme WANG et al., 2015 et KUMAR et GUPTA, 2022 proposent des scénarios hybrides entre la génération d'échantillons et l'échantillonnage de bases de données. Des échantillons artificiels sont d'abord générés puis leurs plus proches voisins dans une base de données existante sont sélectionnés pour la labélisation.

Le problème d'apprentissage

Pour pouvoir appliquer des méthodes d'apprentissage actif à un problème industriel, celui-ci doit être formalisé en un problème d'apprentissage automatique, et plus particulièrement d'apprentissage supervisé. Pour ce faire, il est nécessaire d'identifier les données X pouvant servir d'entrée au modèle d'apprentissage automatique, ainsi que la sortie y qui doit être prédite. On parle aussi de cible ou de label. Le rôle du modèle d'apprentissage automatique, qu'il soit entraîné par apprentissage actif ou passif, est de réaliser une prédiction $\hat{y}$ de y à partir de X. Il est à noter que les données d'entrée X devront, par ailleurs, être mises sous une forme utilisable par le modèle d'apprentissage automatique qui sera sélectionné pour la tâche donnée. Beaucoup de ces modèles, par exemple, requièrent que ces données d'entrée soient mises sous la forme d'un vecteur de caractéristiques de dimension fixe $X \in \mathbf{R}^d$. Par exemple, WEIGL et al., 2016 proposent l'utilisation de l'apprentissage actif pour un problème de détection de défauts à la surface de puces microfluidiques. Les données d'entrée sont d'abord récoltées sur la ligne de fabrication sous forme d'image par une caméra à balayage linéaire. Ces images sont ensuite prétraitées et transformées sous la forme d'un vecteur de caractéristiques de dimension 24 : $X \in \mathbf{R}^{24}$. Cinq types de défauts ont été identifiés par des experts métiers. L'objectif est de prédire la présence d'un de ces cinq types de défaut, ou leur absence. La cible est ici modélisée par un entier entre 1 et 6 : $y \in [\![1; 6]\!]$.

Les problèmes d'apprentissage supervisé sont séparés en problèmes de classification et problèmes de régression. Cette distinction peut avoir un impact sur le choix de la stratégie d'échantillonnage. Certaines stratégies sont, en effet, spécifiques aux problèmes de régression ou de classification.

La distinction entre problèmes de régression et de classification n'est pas toujours évidente en pratique. En général, les problèmes de classification sont des problèmes tels que la cible est à valeur dans un espace discret fini, sans structure particulière. Par exemple, dans le cas de l'exemple de WEIGL et al., 2016, la cible y ne peut prendre que six valeurs. De plus, que le cas "absence de défaut" soit encodé par $y = 1$ ou $y = 6$ n'a aucune importance en pratique. Additionner ou établir

une relation d'ordre entre les différentes valeurs que peut prendre la cible n'a pas non plus de sens en pratique. Il s'agit donc bien d'un problème de classification. Les problèmes de régression sont souvent (mais pas toujours) des problèmes pour lesquels la cible prend une valeur dans un espace continu, souvent un sous-ensemble de $\mathbf{R}$ ou $\mathbf{R}^n$. Par exemple, prédire la consommation électrique d'un bâtiment en fonction de la date et de données météorologiques est un problème de régression.

Le budget

Le budget est, ici, considéré comme une contrainte émise sur la quantité de données pouvant être labélisées par l'*oracle*, soit durant le processus de sélection complet soit par unité de temps. La définition exacte du budget varie d'une application à l'autre, en particulier en fonction du scénario d'échantillonnage. Dans le cadre de l'échantillonnage de base de données, ce budget est généralement considéré comme étant un nombre maximum de points de données pouvant être labélisés par l'*oracle*. Cette définition est cependant inadaptée dans le cas de l'échantillonnage de flux, en particulier lorsque le flux échantillonné est considéré infini. Le budget est alors plutôt considéré comme un taux de données labélisées. Ienco et al., 2013, par exemple, séparent le flux en blocs consécutifs et définit le budget comme la proportion de chaque bloc échantillonné devant être labélisée par l'*oracle*. Kottke, Krempl et Spiliopoulou, 2015 considèrent plutôt le budget comme une valeur cible autour de laquelle doit osciller la proportion d'échantillons labélisés au cours du temps. Plus particulièrement, les auteurs proposent un algorithme qui garantit que la proportion de données labélisées jusqu'à un instant quelconque reste dans un intervalle de tolérance autour de cette valeur cible.

La stratégie d'échantillonnage

De nombreuses stratégies d'échantillonnages ont été proposées dans le cadre de l'apprentissage actif. Ces stratégies peuvent être distinguées entre celles basées sur les modèles (sous-entendu les modèles d'apprentissage automatique entraînés par apprentissage actif, ces méthodes sont dites *model-based* en anglais) et celles non basées sur les modèles, mais utilisant uniquement la structure des données d'entrée (méthodes dites *model-free* en anglais).

Les stratégies basées sur les modèles font intervenir les prédictions réalisées par les modèles en cours d'entraînement dans le processus d'échantillonnage. Ces modèles sont régulièrement réentraînés lorsque suffisamment de nouveaux exemples d'entraînement ont été labélisés, la stratégie est mise à jour et le processus d'échantillonnage continue. Par exemple, de nombreuses stratégies font intervenir une mesure heuristique $u(X, \hat{y})$ de l'utilité de labéliser un point de données. Ces mesures peuvent parfois être interprétées comme des mesures d'un niveau de confiance ou d'incertitude pouvant être associé à la prédiction $\hat{y}$. Les données maximisant (ou minimisant) cette mesure sont alors échantillonnées et labélisées en priorité. Le modèle prédictif est mis à jour à l'aide de ces nouveaux exemples d'entraînement, ce qui amène à un changement des prédictions et de la mesure d'utilité. De nouveaux échantillons sont sélectionnés et ainsi de suite. La figure 2.9 présente le principe d'une stratégie basée sur les modèles dans le cas de l'échantillonnage de flux de données. La prédiction du label par le modèle d'apprentissage automatique est réalisée avant que ne soit prise la décision de labéliser ou non le point de données.

Trois mesures d'utilité populaires pour les problèmes de classification sont l'incertitude, la marge, et

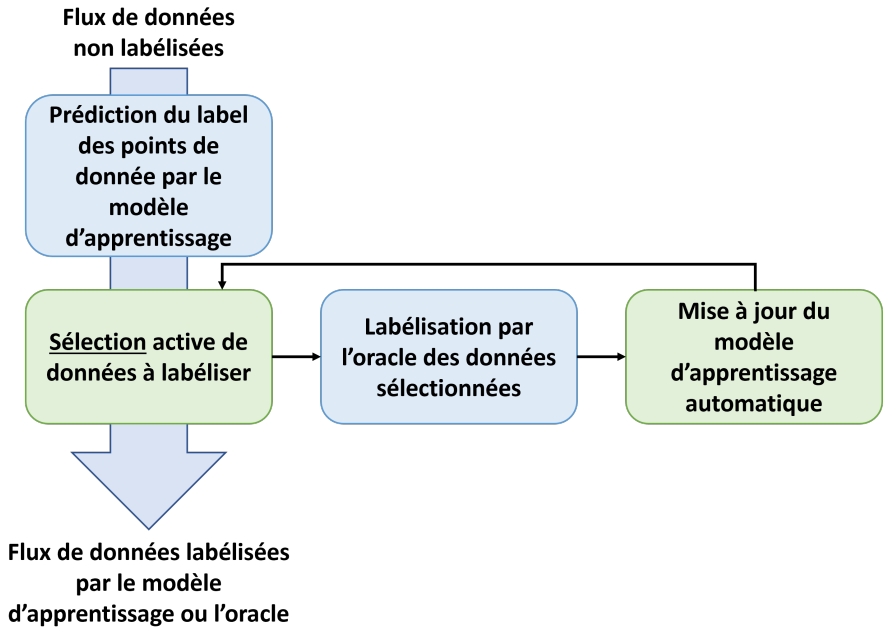

FIGURE 2.9 : Échantillonage actif, basé sur le modèle, dans le cas de l'échantillonnage de flux de données.

l'entropie (SETTLES, 2009). Bien que seule la première de ces mesures soit canoniquement nommée "incertitudes" toutes ces mesures peuvent être interprétées comme des mesures heuristiques de l'incertitude ou de la confiance associées aux prédictions. Ces mesures sont basées sur le fait que beaucoup de modèles de classification ne prédisent pas uniquement une seule classe mais génèrent plutôt, pour chaque classe possible $c \in \mathcal{C}$, une quantité $\hat{y}_c \in [0, 1]$. $\mathcal{C}$ représente ici l'ensemble des classes considérées dans le problème de classification. De plus, $\sum_{c \in \mathcal{C}} \hat{y}_c = 1$. Les prédictions $\hat{y}_c$ sont alors assimilées à la probabilité que le vrai label de X soit c, considérant le modèle de prédiction.

Les mesures d'utilités sont alors :

— Pour l'incertitude, $u(X, \hat{y}) = \hat{y}^1 = max_{c \in \mathcal{C}} \hat{y}_c$. Cette mesure privilégie les points de données pour lesquels la probabilité associée par le modèle à la classe prédite est basse.
— Pour la marge, $u(X, \hat{y}) = \hat{y}^1 - max_{c \in \mathcal{C} \setminus \hat{y}^1} \hat{y}_c$. Cette mesure privilégie les points de données pour lesquels la distinction entre la classe la plus probable et la deuxième classe la plus probable est plus faible.
— Pour l'entropie, $u(X, \hat{y}) = \sum_{c \in \mathcal{C}} \hat{y}_c log(\hat{y}_c)$. Cette mesure privilégie les points de données pour lesquels la distribution de probabilité de chaque classe est proche d'une mesure uniforme.

D'autres mesures d'utilité ont été proposées dans le cadre des problèmes de régressions. Par exemple, BURBIDGE, ROWLAND et KING, 2007 proposent une mesure d'utilité pour le cas où le modèle prédictif est un ensemble de plusieurs modèles. Dans ce cas, la prédiction de l'ensemble est donnée par $\hat{y} = \frac{1}{B} \sum_{i=1}^{B} \hat{y}_{.i}$, avec B le nombre de modèles dans l'ensemble, et $\hat{y}_{.i}$ les prédictions des modèles individuels. Considérant que cette prédiction est la moyenne des prédictions réalisées par

les différents modèles de l'ensemble, les auteurs proposent d'utiliser la variance de ces prédictions comme mesure du niveau de confiance qui peut lui être associée, c'est-à-dire :

$$a(X) = \frac{1}{B-1} \sum_{i=1}^{B} (\hat{y}_{.i}(X) - \hat{y}(X))^2\} \tag{2.1}$$

Cette mesure est nommée ambiguïté par les auteurs. Une variante, $s(x)$ a été proposée par Farache et al., 2022 dans le cas où le modèle est spécifiquement un modèle de forêt aléatoire, dont les différents arbres ont été entraînés sur des échantillons bootstraps de la base d'entraînement complète. Cette mesure est un estimateur de la variance d'échantillonnage de la prédiction, proposé par Wager, Hastie et Efron, 2014. Cette mesure estime le degré de changement de la prédiction si la forêt était réentraînée sur une nouvelle base d'entraînement :

$$s(x) = \sum_{\omega=1}^{\Omega} Cov[J_{i\omega}, \hat{y}_{.}i(x)]^2 + \sum_{\omega=1}^{\Omega} (\hat{y}_{-\omega}(x) - \hat{y}(x))^2 - \frac{e\Omega}{B} a(x) \tag{2.2}$$

Ici, Ω désigne le nombre d'exemples dans la base d'entraînement. J_ω est une variable aléatoire comptant combien de fois le ω^e point de données est présent dans la base d'entraînement de l'arbre $\hat{y}_{.}$. $Cov[J_\omega, \hat{y}_{.}(x)]$ est l'estimateur de covariance classique entre cette variable et la prédiction d'un arbre entraîné sur la base bootstrap associée. $\hat{y}_{-\omega}$ est la forêt constituée de tous les arbres pour lesquels le ω^e point de données n'est pas dans la base d'entraînement. $a(X)$ est la mesure d'ambiguïté.

Les stratégies non basées sur les modèles se fondent, au contraire, uniquement sur la géométrie et la distribution des points de données X. Elles ont l'avantage de pouvoir être utilisées sur n'importe quel modèle prédictif, qu'il soit un modèle de régression ou de classification. De plus, dans le cas du scénario d'échantillonnage de base de données, elles limitent le besoin de réentraîner plusieurs fois le modèle prédictif. Par exemple, Baram, Yaniv et Luz, 2004 proposent une stratégie sélectionnant itérativement les points de données les plus éloignés de la base de données précédemment sélectionnée. Dans ce cas, la mesure d'utilité utilisée pour sélectionner les points de données à labeliser est :

$$diss(X) = \min_{x_i in \mathcal{D}} d(X, x_i)\,, \tag{2.3}$$

avec $\mathcal{D}$ l'ensemble des données déjà sélectionnées pour labélisation et d une distance.

La figure 2.10 présente le principe d'une stratégie basée sur les modèles dans le cas de l'échantillonnage de flux de données. La prédiction du label par le modèle d'apprentissage automatique est réalisée, si nécessaire, après que soit prise la décision de labéliser ou non le point de données.

D'autres stratégies, proposées dans la littérature, privilégient les points présents dans des régions denses de l'espace des caractéristiques ou tirent parti de la structure des données en clusters (Wang et al., 2017).

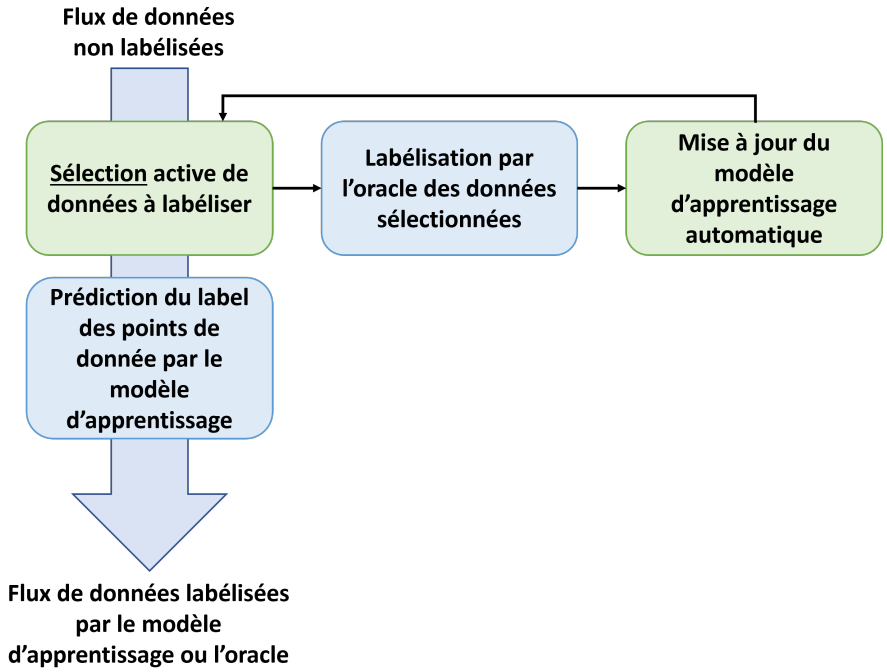

FIGURE 2.10 : Échantillonage actif, non basé sur le modèle, dans le cas de l'échantillonnage de flux de données.

Apprentissage automatique et ombres numériques

Les stratégies proposées dans la littérature sur l'apprentissage actif apparaissent utiles dans le cadre des jumeaux numériques qui doivent améliorer et adapter leurs modèles de manière efficiente à partir de grandes masses de données, par exemple pour l'entraînement de modèles substituts. De plus, les trois scénarios d'échantillonnage présentent un intérêt pour différents cas d'usage et types d'ombres et jumeaux numériques.

Les scénarios d'échantillonnage de base de données peuvent, par exemple, être utilisés dans le cadre des jumeaux numériques prototypes, non encore reliés à un homologue physique, pour construire des métamodèles de modèles de simulation. Par exemple, ASADI et al., 2021 proposent une application pour l'optimisation d'opérations de soudage. Le nombre de configurations de soudage est fini et connu, mais trop large pour être intégralement exploré par un jumeau numérique prototype basé uniquement sur des simulations par éléments finis. Les auteurs proposent donc une stratégie d'apprentissage actif pour entraîner un métamodèle de ce modèle de simulation pour prédire la distorsion de la pièce soudée pour différentes configurations.

Les scénarios d'échantillonnage de flux peuvent être utilisés par des ombres et jumeaux numériques qui collectent en continu de grandes masses de données sur leur homologue physique. Ces données peuvent être utilisées pour l'amélioration ou l'adaptation des modèles numériques basés sur ces données. Lorsque ces données doivent être labélisées par des experts extérieurs au jumeau ou des modèles de simulation complexes, l'apprentissage actif peut être utilisé. GARDNER et al., 2020, par exemple, proposent une étude appliquant l'apprentissage actif pour l'adaptation d'un

jumeau numérique d'une structure en aluminium à des changements abrupts dans la structure physique. L'ombre numérique est constituée de deux modèles. Le premier modèle est un système d'équations différentielles linéaires basées sur une modélisation de la physique de la structure, à partir de la connaissance des experts impliqués dans son développement. Ce modèle réalise une première estimation de l'accélération du dernier étage de la structure lorsque celle-ci est soumise à des vibrations extérieures. L'estimation de ce premier modèle est ensuite affinée par un modèle d'apprentissage automatique entraîné sur des données collectées en continu sur le jumeau physique. Ce deuxième modèle permet notamment au jumeau de s'adapter à un changement abrupt, non pris en compte par le modèle physique. Un objectif secondaire évoqué par les auteurs est de prévenir les utilisateurs lorsque le jumeau anticipe que certaines prédictions sont de faible qualité.

L'apprentissage actif a cependant été peu étudié dans le cadre des ombres et jumeaux numériques. Pour illustration, la recherche ("Digital twin" OR "Digital shadow") AND "Active learning" a été réalisée en février 2023 dans les bases scientifiques "Web of Science", scopus et IEEE. Les mots-clés sont cherchés uniquement dans les titres, résumés et mots-clés. Les articles publiés par Sylvain Chabanet (l'auteur de ce mémoire), ceux traitant du concept d'apprentissage actif en science de l'éducation et non en apprentissage automatique, ainsi que les revues de littérature ont été retirés. Sur ces bases, seuls deux documents correspondent aux critères décrits : Gardner et al., 2020 et Hughes et al., 2022, ce qui est très peu.

2.3.3 Détection de la dérive conceptuelle et adaptation des modèles

Comme énoncé dans la section 2.2, la capacité des ombres et jumeaux numériques à suivre leur homologue physique sur des temps longs à l'échelle de son cycle de vie est un aspect important de ces concepts. Cela nécessite cependant que les modèles inclus dans les ombres et jumeaux numériques doivent être capables de détecter et de s'adapter à des changements dans la structure ou l'environnement de l'homologue physique. De tels changements pourront par exemple avoir lieu durant la phase de service d'ombres numériques de produits, ou de production pour les ombres numériques de processus. Ils peuvent être causés par de l'usure, des pannes, ou des changements dans la matière première utilisée.

Notion de dérive conceptuelle

Dans la littérature relative à l'apprentissage automatique en flux de données, ces changements sont appelés dérives conceptuelles. Ces dérives sont formalisées mathématiquement par un changement au cours du temps t dans la loi de probabilité jointe entre les données d'entrée X et les cibles y, $\mathbf{P}_t(X, y)$. Une taxonomie très complète de différents types de dérive conceptuelle est présentée dans Webb et al., 2016.

Les dérives conceptuelles peuvent tout d'abord être classées en dérives conceptuelles dites réelles et dérives conceptuelles dites virtuelles. Tout d'abord rappelons que la loi de probabilité jointe entre X et y peut être décomposée en $\mathbf{P}_t(X, y) = \mathbf{P}_t(y|X)\mathbf{P}_t(X)$.

— Lorsque les changements concernent la loi de probabilité des données d'entrée X, c'est-à-dire lorsqu'il existe deux temps t et u tels que $\mathbf{P}_t(X) \neq \mathbf{P}_u(X)$, on parle de dérive conceptuelle virtuelle. Ce types de dérive concerne donc les changements dans la fréquence d'apparition de certains types de données, voir l'apparition de cas jamais observés auparavant. Cependant, le lien entre les données d'entrée et leur cible reste identique au cours du temps.

— Lorsque les changements concernent la loi de probabilité conditionnelle liant les cibles y aux données d'entrée X, c'est-à-dire lorsqu'il existe deux temps t et u tels que $\mathbf{P}_t(y|X) \neq \mathbf{P}_u(y|X)$, on parle de dérive conceptuelle réelle. Dans ce cas la relation entre les données d'entrée et la cible change au cours de temps. Cela peut, par exemple, être dû à l'usure de pièces ou à des changements de réglage. Cela signifie également que les données collectées avant la dérive ne sont plus représentatives du comportement de l'homologue physique et que les modèles entraînés sur ces données verront leurs performances diminuer.

Les dérives virtuelles peuvent également être distinguées par la rapidité des changements en cause. En particulier, comme illustré figure 2.11, le changement peut être abrupt ou incrémental.

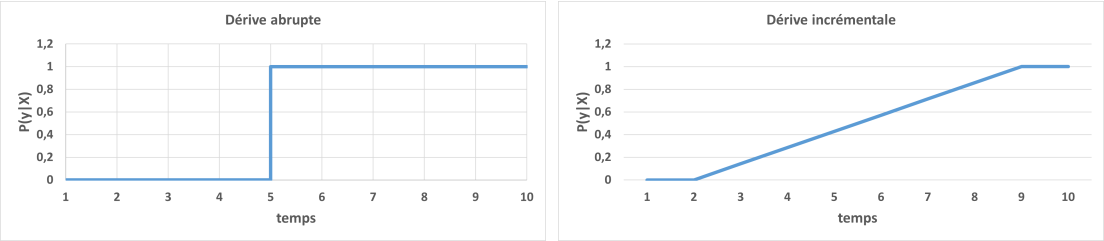

FIGURE 2.11 : Exemples de dérives abrupte (à gauche) et incrémentale (à droite)

Détection de dérives conceptuelles

Beaucoup d'algorithmes utilisent notamment une mesure de l'erreur commise à chaque prédiction. Contrôler ce flux d'erreurs permet de détecter à la fois des dérives virtuelles et réelles. De plus, une alerte n'est lancée que lorsque cette dérive est dommageable aux performances du modèle prédictif. Une hypothèse sous-jacente très forte utilisée par de nombreux auteurs pour tester ces méthodes est cependant que les vraies valeurs y des cibles associées à tous les points de données générés par le flux sont disponibles immédiatement après que la prédiction de cette cible ait été faite pour en évaluer l'erreur. Dans le cas où ces vraies valeurs ne sont pas accessibles, d'autres auteurs proposent de contrôler le flux des caractéristiques (REIS et al., 2016), ou un flux composé de mesure d'incertitude de chaque prédiction (BAIER et al., 2021). La suite de cette section présente, en particulier, quatre méthodes de détection de dérives identifiées pour leur popularité dans la littérature, la diversité des concepts sur lesquels elles sont basées, et le relativement faible nombre d'hypothèses nécessaires pour les utiliser.

Par exemple, l'algorithme de Page-Hinkley est utilisé pour la détection de dérives dans GAMA, SEBASTIAO et RODRIGUES, 2013. Cet algorithme repose sur le test de Page-Hinkley introduit par PAGE, 1954 dans le contexte du contrôle statistique de la production. Il est conçu pour détecter des changements dans la moyenne d'un processus gaussien (x_t). Ce test calcule pour chaque nouvelle arrivée T la quantité :

$$m_T = \sum_{t=1}^{T} (x_t - \bar{x}_T - \delta)\,, \tag{2.4}$$

avec $\bar{x}_T$ la moyenne de la série jusqu'à l'instant T. Le minimum M_T des m_t jusqu'à l'instant T est également mesuré, et une alerte est lancée si $m_T - M_T > h$ avec h un seuil fixé par l'utilisateur. De même, le facteur δ peut être interprété comme le niveau de dérive maximum toléré par l'utilisateur. Il pourra être noté que cette version du test de Page-Hinkley ne détectera que des augmentations dans la moyenne de la série (x_t). Gomes et al., 2019 proposent d'utiliser pour x_t différentes estimations de l'erreur préquentielle des prédictions, c'est à dire de la moyenne de toutes les erreurs commises par le modèle prédictif jusqu'à l'instant t, $x_t = \frac{1}{t}\sum_{i=1}^{t} \epsilon_i$, avec ϵ_i l'erreur de prédiction pour le i^e point de données. Un avantage de cette méthode est qu'elle permet une estimation du temps de début de la dérive comme étant le temps d'atteinte du minimum M_T, à condition que le facteur δ soit suffisamment important (Hinkley, 1971).

Une deuxième méthode de détection de dérives inspirée du contrôle statistique des processus est basée sur l'utilisation de cartes de contrôle. L'avantage des cartes de contrôle est qu'il en existe de nombreux types permettant la surveillance de diverses caractéristiques, telles que la moyenne et l'écart-type. Les cartes de contrôle sont des outils graphiques. Elles sont souvent constituées d'une ligne centrale, d'une ligne de contrôle supérieure et d'une ligne de contrôle inférieure. Dans certains cas, on ajoute les lignes supérieure et inférieure de surveillance, plus resserrées autour de la ligne centrale. Des échantillons de valeurs sont collectés à intervalles réguliers sur la série x_t et utilisés pour calculer quelques statistiques représentatives telles que la moyenne et l'écart-type. Ces statistiques sont alors reportées sur les cartes de contrôle sous la forme de séries temporelles. Une alarme est lancée lorsqu'une configuration inhabituelle de ces points est observée, typiquement lorsqu'ils sortent des lignes de contrôle. Une hypothèse sous-jacente est cependant que les erreurs sont au moins approximativement distribuées suivant une loi normale. De nombreux types de cartes de contrôle on été proposées par différents auteurs pour la détection des dérives. Noyel et al., 2016, par exemple, utilisent une carte np pour surveiller la proportion d'erreurs de classification. Noyel et al., 2013 utilisent une carte S pour contrôler l'écart-type de l'erreur de prédiction. Ross et al., 2012 utilisent une carte de contrôle à moyenne mobile avec pondération exponentielle (Exponentially Weighted Moving Average, EWMA) pour contrôler le taux d'erreurs de classification.

D'autres stratégies de détection ont été introduites dans l'optique de surveiller non pas la stationnarité des erreurs commises par le modèle mais celle des données d'entrée. L'intérêt de ces méthodes est de ne pas recourir aux cibles y. C'est par exemple le cas de la méthode proposée par Reis et al., 2016, qui l'utilise pour la détection de dérives virtuelles en surveillant les distributions des vecteurs de caractéristiques. Rien n'empêche cependant de l'utiliser pour la détection de dérives réelles en surveillant par exemple l'erreur commise par un modèle de régression. Cette méthode se base sur le test de Kolmogorov-Smirnov qui est un test non paramétrique testant l'hypothèse que deux échantillons de données proviennent de la même distribution statistique. L'avantage de ce test est donc qu'il ne fait aucune hypothèse sur la distribution statistique des échantillons comparés. La statistique du test à 2 échantillons est :

$$D_{ks} = \sup_t\{|F_1(t) - F_2(t)|\}\,, \tag{2.5}$$

avec F_1 la fonction de répartition empirique du premier échantillon et F_2 la fonction de répartition

empirique du deuxième.

Pour la détection de dérives, ce test est appliqué de façon répétée pour comparer une fenêtre de référence, fixe, et une fenêtre mobile rassemblant les dernières valeurs observées. Une alarme est déclenchée si la p-value de ce test est inférieure à un seuil fixé par l'utilisateur. Une alternative au test de kolmogorov-smirnov est le test d'Anderson-Darling à k échantillons. Il s'agit également d'un test non-paramétrique testant l'égalité des distributions de plusieurs échantillons. La statistique du test à k échantillons peut être écrite comme :

$$A_k^2 = \sum_{i=1}^{k} n_i \int_U \frac{(F_i(t) - H(t))^2}{H(t)(1 - H(t))} dH(t)\,, \tag{2.6}$$

Avec n_i la taille du i^e échantillon, $H = \frac{\sum_i n_i F_i}{N}$ avec $N = \sum_i n_i$, et $U = \{t \in \mathbf{R} | H(t) < 1\}$. Contrairement au test de Kolmogorov-Smirnov qui se concentre sur les différences au centre des distributions, le test d'Anderson-Darling donne plus de poids aux queues des distributions (Tan, Lee et Salehi, 2022). Il a été montré empiriquement que ce test est plus puissant que le test de Kolmogorov-Smirnov dans le cadre spécifique des tests de normalité (Razali, Wah et al., 2011). Cependant, le calcul des p-values du test d'Anderson-Darling à k échantillons nécessite l'interpolation de valeurs tabulées par méthodes de Monte-Carlo, ce qui limite en pratique l'étendu du seuil de test utilisable.

La table 2.1 présente les avantages et inconvénients respectifs des méthodes évoquées.

Adaptation à la dérive conceptuelle

Les modèles utilisés pour l'apprentissage automatique à partir de flux de données changeant peuvent être classés en "*batch-incremental*" et "*instance-incremental*" (Read et al., 2012).

Les modèles dit *instance-incremental* sont mis à jour dès qu'un nouveau point de données est disponible. Un modèle pouvant être implémenté pour apprendre de chaque nouvelle donnée disponible est par exemple le classifieur bayesien naïf. Ce modèle se base sur un certain nombre de grandeurs statistiques résumant les données observées, par exemple les valeurs moyennes et variances des différentes caractéristiques conditionnées à la classe des données, $\mathbf{E}(X|y)$ et $\mathbf{Var}(X|y)$. Les estimations de ces grandeurs peuvent facilement être mises à jour lorsque de nouvelles données et labels sont générés. Un autre exemple de modèle développé spécialement pour le cas des flux de données massifs sont les arbres de Hoeffding (Domingos et Hulten, 2000). Plus précisément, ils sont développés pour des flux de données massivement labélisées. En tant que modèles d'apprentissage supervisé, les vraies valeurs des cibles y associées aux points de données X générés par le flux sont donc nécessaire à leur entraînement. A chaque nouvelle arrivée, les arbres de Hoeffding réalisent notamment un test, basé sur l'inégalité de Hoeffding, qui décide de réaliser ou non la séparation d'une feuille de l'arbre en deux. Le test réalisé est indépendant de la distribution des données mais est très conservateur, ce qui force l'arbre à ne grandir que très lentement à l'arrivée de nouvelles données labélisées.

Les modèles dit *batch-incrémental* ne sont, eux, mis à jour ou réentraînés que lorsqu'une certaine quantité de nouvelles données a été collectée. D'autres conditions peuvent également être utilisées pour déclencher le réentraînement de ces modèles. Par exemple, Noyel et al., 2016 utilisent un

Méthode	Avantages	Inconvénients
Page-Hinkley (Hinkley, 1971)	— Peut estimer le temps de début de dérive. — Faible coût en mémoire et calcul à chaque nouvelle arrivée. — Différencie entre des tendances croissantes ou décroissantes de la dérive.	— Paramètres difficiles à fixer. — Est surtout adapté aux dérives abruptes des moyennes des distributions. — Ne détecte que des dérives sur la moyenne du flux contrôlé.
Kolmogorov-Smirnov (Reis et al., 2016)	— test statistique non paramétrique, sur les distributions empiriques dans leur ensemble. — Une version incrémentale moins coûteuse en calculs que la simple répétition du test existe.	— Très sensible au seuil fixé du test. — Doit garder deux fenêtres de données en mémoire.
Anderson-Darling (Tan, Lee et Salehi, 2022)	— test statistique non paramétrique, sur les distributions empiriques dans leur ensemble. — Une version incrémentale moins coûteuse en calculs que la simple répétition du test est proposée (annexe A).	— Très sensible au seuil fixé du test. — Doit garder deux fenêtres de données en mémoire. — La distribution statistique du test doit être tabulée empiriquement ce qui limite les seuils d'acceptation utilisable.
Cartes de contrôle (Noyel et al., 2016)	— Faible coût en mémoire et calcul à chaque nouvelle arrivée. — Possibilité de contrôler plusieurs paramètres des flux de données suivant les cas : moyennes, écart-types, proportions...	— Les paramètres des cartes sont souvent dimensionnés suivant des hypothèses strictes sur la distribution des données, notamment de normalité.

Table 2.1 : Avantages et inconvénients de méthodes de détection de dérives.

ANN batch-incrémental pour l'apprentissage en flux en présence de dérives conceptuelles. Un premier ANN est entraîné hors ligne avant le début du flux. Ce modèle n'est mis à jour que lorsqu'une dérive est détectée dans le flux. Cet évènement déclenche la collecte d'une nouvelle base de données d'entraînement qui est utilisée pour réentraîner le modèle. Une stratégie similaire est présentée dans Reis et al., 2016 pour trois types de modèles : un algorithme des plus proches voisins, un arbre de décision et un classifieur bayésien naïf.

Cas des ombres numériques

> L'intégration de méthodes de détection de dérives conceptuelles apparaît comme un élément important pour le développement d'ombres numériques liées à leur homologue physique tout au long de leur cycle de vie. Durant leur phase d'utilisation et de service en particulier, cet homologue physique pourra être sujet à des changements dans ses comportements internes ou son environnement. Ces méthodes de détection pourront alors être utilisées pour avertir l'utilisateur d'une désynchronisation entre les entités physique et virtuelle, menant à une possible baisse de performance des modèles inclus dans l'entité numérique. Il convient également de permettre à ces modèles de bénéficier des données collectées à partir de l'homologue physique pour s'améliorer en continu et réagir aux dérives conceptuelles.

Ces principes sont par exemple évoqués par Gardner et al., 2020 qui présente une stratégie d'apprentissage actif pour l'échantillonnage de flux de données et le développement d'un jumeau numérique d'une structure en aluminium. La stratégie qu'ils proposent n'implémente pas de méthode explicite de détection de dérives. Cependant, le flux est échantillonné suivant une mesure de l'incertitude estimée du modèle dans sa prédiction. La dérive conceptuelle introduite par un changement dans la structure de l'homologue physique se caractérise par l'apparition soudaine de nombreuses données pour lesquelles cette incertitude est élevée, et par un sur-échantillonnage temporaire du flux pour permettre une adaptation rapide des modèles prédictifs. Une application pour une ombre numérique prédisant la consommation électrique de machinerie industrielle a également été proposée par Bermeo-Ayerbe, Ocampo-Martinez et Diaz-Rozo, 2022. Une stratégie de détection de dérives basée sur le test de Page-Hinkley est proposée pour détecter des changements de profils de consommation dû à l'usure et réentrainer un modèle prédictif.

2.3.4 Questions de recherche étudiées durant cette thèse

Comme évoqués précédemment, de nombreux verrous scientifiques sont associés au développement d'O&JN dans l'industrie. Nous nous concentrerons dans la suite de ce mémoire de thèse sur les verrous liés à la coordination/synchronisation de plusieurs modèles numériques réalisant des tâches de prédiction. Nous nous concentrerons, en particulier, sur le cas incluant deux modèles de nature différentes. Le premier est un modèle de simulation, considéré comme ayant un haut niveau de fidélité mais un coût en calcul trop important pour être utilisé seul pour prédire les valeurs cibles associées à tous les points de données collectés auprès du jumeau physique et transmis en flux. Un modèle de substitution basé sur des méthodes d'apprentissage automatique est donc introduit. Ce modèle de substitution reste cependant une approximation du modèle de simulation initial, et il convient de coupler les deux modèles au sein de l'O&JN pour tirer partie de leurs avantages respectif.

La question de recherche principale traitée ici est donc "Comment coupler de manière efficiente ces deux modèles dans la logique des O&JN pour tirer autant que possible parti de leurs avantages respectifs ?". Pour répondre à cette question, une stratégie de couplage entre ces deux modèles est proposée dans le chapitre 3. Cette stratégie s'appuie, notamment, sur la mise en place d'un mécanisme décidant, pour chaque point de données transmit par un flux, si le modèle de simulation doit être utilisé ou si le modèle de substitution est suffisant. L'usage d'une telle stratégie lève ses propres questions scientifiques spécifiques, parmis lesquelles :

— Comment utiliser de manière efficiente les ressources du modèle de simulation ?
— Comment évaluer l'utilité d'utiliser le modèle de simulation au lieu de son modèle de substitution pour un point de données spécifique, et comment utiliser ces mesures d'utilité pour coordonner les deux modèles ?
— Comment intégrer à la stratégie proposée la possibilité de détecter la présence de changements dans le flux de données d'entrée pour réagir en conséquence ?

2.4 Conclusion

Ce chapitre introduit le concept d'ombre numérique, qui dérive de celui de jumeau numérique. La littérature sur les O&JN est très variée et des travaux portant sur des entités numériques très différentes ont été publiés. Ces différences existent tant dans les objectifs et services supportés que dans le domaine d'application.

> Les travaux présentés dans ce mémoire s'inscrivent dans la vision des ombres numériques comme une évolution des systèmes de simulation pour élargir leur utilisation dans l'aide à la décision des niveaux tactique et stratégique au niveau opérationnel. Plus particulièrement, l'application développée considère une instance d'ombre numérique de processus. L'objectif de cette ombre numérique de processus est de réaliser une prédiction, par exemple un état futur, pour chaque élément d'un flux de données. En particulier, ce flux de données considérera la transmission des informations sur les ressources ou produits transitant dans le processus.

Le développement de ce type d'ombre numérique présente, cependant, des verrous scientifiques concernant la rapidité des simulations et l'efficience de l'utilisation des ressources numériques. Comme montré par les éléments de littérature présentés dans ce chapitre, l'utilisation de modèles substituts est une solution intéressante pour pallier à des temps de simulation trop longs. L'utilisation de ces méthodes introduit, cependant, des difficultés propres, que ce soit pour l'entraînement, l'amélioration continue et l'adaptation de ces modèles en cas de dérives. Cela pose également la question de la confiance pouvant être attribuée aux différentes prédictions d'un modèle d'apprentissage automatique, dont les performances sont très dépendantes de la quantité et de la qualité des données utilisées pour l'entraîner.

Pour répondre à ces verrous, le chapitre 3 proposera une stratégie basée sur l'apprentissage actif, ainsi qu'un modèle UML pour son implémentation et son évaluation. Le scénario d'échantillonnage considéré sera donc l'échantillonnage de flux de données. Cette stratégie sera d'abord évaluée sur des flux de données stationnaires et non stationnaires provenant du dépôt UCI. Elle sera, ensuite, appliquée à un jeu de données de résultats de simulation de sciage originaire de la filière forêt-bois canadienne. Une présentation des domaines de l'apprentissage automatique sur lesquels sont basés la stratégie proposée a été effectuée dans ce chapitre, et leur lien avec le concept des O&JN a été investigué.

Chapitre 3

Proposition d'une stratégie de couplage entre un modèle de simulation et son métamodèle d'apprentissage automatique dans le contexte des ombres numériques.

3.1 Introduction

Ce chapitre introduit et détaille la contribution centrale de cette thèse, c'est-à-dire la proposition d'une stratégie de couplage entre un modèle de simulation et un modèle d'apprentissage automatique pour le développement d'O&JN. Il ne s'agit en aucun cas de développer une ombre numérique finalisée, mais d'en étudier des éléments précis et de proposer une preuve de concept de la stratégie proposée. En particulier, la stratégie de couplage développée ici suppose l'existence préalable de deux éléments constitutifs de l'ombre numérique. Le premier est un modèle de simulation ayant un haut niveau de fidélité, mais étant trop lent pour être utilisé sur la totalité des points de données collectés au cours du temps depuis l'homologue physique. Le deuxième élément, un modèle d'apprentissage automatique, est donc introduit pour réaliser cette même tâche de prédiction. Cependant, comme présenté dans le chapitre 2, ces modèles présentent leurs propres inconvénients, notamment leur difficulté à généraliser face à des données trop différentes de leurs bases d'entraînement, par exemple en cas de dérive conceptuelle virtuelle ou réelle.

La stratégie de couplage proposée a, en particulier, les objectifs suivants :

— Une utilisation efficiente des ressources en calculs, et notamment des ressources de simulation.
— L'amélioration continue du modèle d'apprentissage automatique au fur et à mesure que de nouveaux points de données sont disponibles, les erreurs commises par le modèle étant évaluées par une fonction de perte $l(\hat{y}, y)$ mesurant le coût causé par le fait de réaliser une

prédiction $\hat{y}$ pour un point de données dont le vrai label est y.

— La détection de dérives conceptuelles pour la surveillance du système et l'aide à la décision.

La suite de ce chapitre est organisée comme suit. D'abord, la section 3.2 présente le principe général de la stratégie proposée. Elle est détaillée dans la section 3.3 à l'aide de diagrammes UML. Des éléments de l'environnement expérimental seront également présentés. Une analyse du comportement de la stratégie proposée en régime stationnaire est présentée dans la section 3.4. Enfin, la section 3.5 présente des premiers résultats expérimentaux obtenus sur huit jeux de données UCI pour évaluer plusieurs variantes de la stratégie proposée et les comparer à une stratégie de référence.

3.2 Principe et système étudiés

La stratégie de couplage proposée considère l'existence d'un flux de données transmettant de nouveaux points de données à des *instants aléatoires*. Pour chaque point de données transmis par le flux, une cible y doit être prédite. Cette prédiction peut être réalisée soit par un modèle de simulation soit par un modèle d'apprentissage automatique entraîné à la même tâche de prédiction. Le temps moyen entre les arrivées de deux points de données est considéré trop court pour que le modèle de simulation puisse être utilisé sur l'intégralité des points de données générés. Tout ou partie des prédictions doivent donc être réalisées par le modèle d'apprentissage automatique. Le modèle de simulation est supposé posséder un haut niveau de fidélité mais être, en contrepartie, lent à exécuter. Nous supposons que plusieurs instances de simulation peuvent être lancées en parallèle. En reprenant la terminologie de la théorie des files d'attentes, nous dirons avoir plusieurs *serveurs* pouvant réaliser les simulations. Une borne supérieure au nombre d'instances pouvant être lancées en parallèle est cependant fixée, motivée en pratique par des contraintes matérielles. Cette borne est notée N_{max} dans la suite de ce mémoire. Le cas simplifié considérant un nombre infini de serveurs a, néanmoins, été étudié et a mené à plusieurs communications en conférences internationales : Chabanet, El-Haouzi et Thomas, 2022 ; Chabanet, El Haouzi et Thomas, 2023 ; Chabanet et al., 2023. Le modèle d'apprentissage automatique est conçu comme un métamodèle du modèle de simulation, entraîné par apprentissage supervisé. En particulier, il n'est entraîné que sur des données tirées du flux de données et labélisées par le simulateur. Aucune autre source de données labélisées n'est considérée.

Le métamodèle n'étant qu'une approximation du modèle de simulation, l'utiliser pour prédire les labels de tous les points de données transmis par le flux induit des prédictions de qualité moindre en moyenne qu'en réalisant certaines de ces prédictions avec le modèle de simulation. Lorsqu'un nouveau point de données est généré par le flux, il est donc nécessaire de décider si le modèle de simulation doit être utilisé ou non. Puisque le modèle de simulation ne peut pas être utilisé pour la totalité des points de données transmis par le flux, il est fait le choix de ne pas créer de file d'attente devant le modèle de simulation. Ainsi, si le nombre d'instances de simulation tournant à l'instant t, N_t, est égal à la borne N_{max}, seul le modèle d'apprentissage automatique peut être utilisé.

Une stratégie naïve pour décider des labels de quels points de données prédire avec quel modèle est de lancer une simulation dès qu'un nouveau point est généré par le flux et qu'il est matériellement possible de lancer une simulation, c'est-à-dire, si $N_t < N_{max}$. Cette méthode revient à maximiser le taux d'utilisation du simulateur et nous servira de méthode de référence par la suite, sous le nom

de *maximisation naïve*. Cependant elle ne prend pas en compte le fait que les prédictions réalisées par le modèle d'apprentissage automatique peuvent se voir attribuer des niveaux de confiance, ou d'incertitude, différents. En supposant que ces mesures soient corrélées avec l'erreur commise par le modèle d'apprentissage automatique, il est souhaitable d'utiliser les ressources de simulation pour les points de données pour lesquels la prédiction faite par le modèle d'apprentissage automatique est la plus incertaine. Cela requiert, cependant, de ne pas saturer les serveurs de simulation comme le fait la stratégie *maximisation naïve*, pour gagner en flexibilité dans la prise de décision.

De telles mesures, souvent heuristiques, associant un niveau d'incertitude aux prédictions faites par des modèles d'apprentissage automatique sont souvent utilisées dans le cadre de l'apprentissage actif. Plusieurs mesures ont été présentées dans le chapitre 2. Bien que l'échantillonnage de flux de données dans le cadre de l'apprentissage actif ait pour but principal l'amélioration continue des performances du modèle d'apprentissage automatique, beaucoup de mesures développées dans le cadre de l'apprentissage actif apparaissent comme de bons candidats dans le contexte de cette étude pour identifier les points de données dont la prédiction par le modèle de substitution est la plus incertaine.

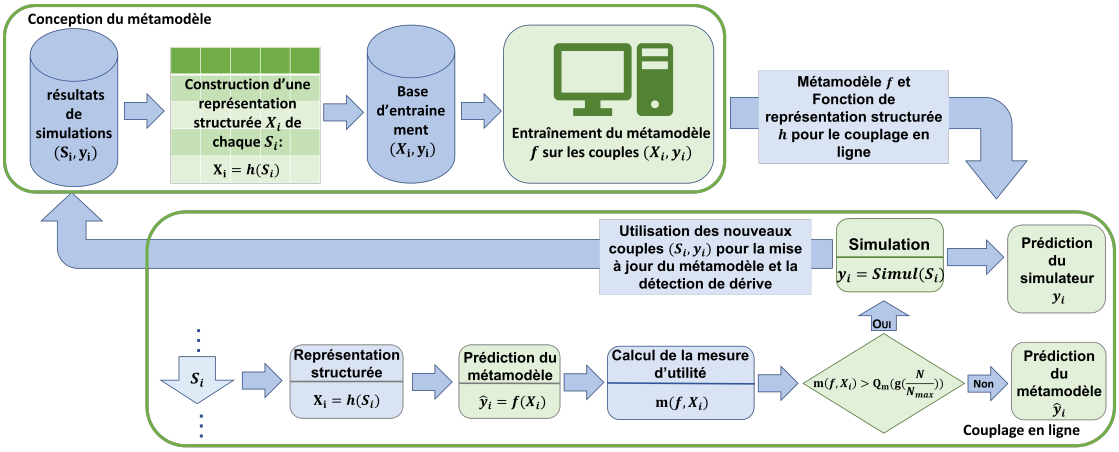

FIGURE 3.1 : Principe général de la stratégie de couplage proposée

Le principe général de la stratégie proposée est présenté figure 3.1. Elle est séparée en deux parties. La première est la conception et l'entraînement initial du modèle d'apprentissage automatique à partir d'une base initiale de résultats de simulation. Cet apprentissage est réalisé hors ligne. Lorsqu'ils seront considérés, les réentraînements de ce modèle seront également réalisés hors ligne. Les structures de données prises en compte par un modèle de simulation peuvent être différentes de celles devant être données en entrée du modèle d'apprentissage automatique. Dans la suite de ce mémoire, nous noterons S_i les données d'entrée du modèle de simulation. Ces données ne sont pas nécessairement exploitables directement par le modèle d'apprentissage automatique. Nous distinguerons donc $X_i = h(S_i)$ les données d'entrée du modèle d'apprentissage automatique, obtenues par une transformation h appliquée aux données d'entrée du modèle de simulation. Une première étape de la conception du modèle d'apprentissage automatique est donc la conception ou la sélection de la fonction h permettant la structuration des S_i en X_i. Il s'agit de l'étape d'ingénierie des caractéristiques (*features engineering* en anglais). La fonction h peut simplement être la fonction identité ou intégrer des transformations plus complexes des S_i. De même, un algorithme d'apprentissage automatique adapté au type de caractéristiques utilisé et au problème d'apprentissage considéré doit être sélectionné, puis entraîné sur les couples (X_i, y_i). Le modèle

d'apprentissage automatique ainsi obtenu peut alors être utilisé en ligne, en conjonction avec le modèle de simulation dont il est le métamodèle.

Le couplage en ligne se déroule comme suit. Un flux transmet, à intervalle de temps aléatoire, des points de données constitués des données d'entrée du modèle de simulation, S_i. Ces données d'entrée sont transformées par le biais de la fonction h pour être exploitable par le modèle d'apprentissage automatique f. Une première prédiction $\hat{y} = f(X_i)$ de la cible est réalisée par le modèle d'apprentissage automatique. Une mesure du niveau d'incertitude, $m(X_i, f)$, est alors réalisée. Cette mesure est utilisée pour prendre la décision de réaliser ou non une simulation pour ce point de données suivant un mécanisme qui sera détaillé par la suite. S'il est décidé de lancer une simulation, le point de données sera labélisé par le simulateur. Le résultat de simulation pourra remplacer la prédiction du modèle d'apprentissage s'il arrive à temps, et dans tous les cas pourra être ajouté à la base d'entraînement labélisée pour améliorer les performances du modèle d'apprentissage automatique.

La décision de lancer une simulation sera prise si $m(X, f)$ est supérieur à un seuil dépendant du taux de remplissage des serveurs de simulation, c'est à dire si :

$$m(X, f) > Q_m(g(N))\,, \tag{3.1}$$

avec Q_m la fonction quantile de la variable aléatoire $m(X, f)$, c'est à dire la fonction telle que $Q_m(z) = \inf_{\xi \in \mathbf{R}}\{\xi | \mathbf{P}(m(X, f) \leq z) \geq \xi\}$. Cette fonction peut facilement être estimée sur une fenêtre glissante sur le flux de données. De même, g est une fonction de $[\![0, N_{max}]\!]$ dans $[0, 1]$ et $N = N_t$ est le nombre de simulations en cours d'exécution à l'instant t ou doit être prise la décision de lancer ou non une nouvelle simulation. Le seul prérequis pour g est que $g(N_{max}) = 1$ de sorte que, par convention, $Q_m(g(N_{max}))$ puisse se voir attribuer la valeur $+\infty$ et que la condition 3.1 ne soit jamais vérifiée si tous les serveurs sont occupés. Deux cas particuliers sont considérés.

Le premier est équivalent à la stratégie de maximisation naïve :

$$g(N) = \begin{cases} 0 \,\text{si}\, N < N_{max} \\ 1 \,\text{sinon} \end{cases} \tag{3.2}$$

Le deuxième considère $g(N)$ comme une fonction affine de N :

$$g(N) = \frac{N}{N_{max}}\,. \tag{3.3}$$

Ainsi, le seuil de décision sur la mesure d'utilité d'une prédiction requis pour lancer une simulation est d'autant plus haut que peu de serveurs sont disponibles.

Dans le système décrit ici, plusieurs flux de données peuvent être surveillés pour la détection de dérives conceptuelles. Ces flux ont des avantages et des inconvénients différents, et le choix du flux surveillé devra dépendre du cas d'usage et du type de dérive attendu.

— Le premier flux possible est celui des caractéristiques, c'est-à-dire celui des X_i. C'est ce flux qui est utilisé, par exemple, par Reis et al., 2016. L'avantage principal de ce flux est que

de nouvelles valeurs sont disponibles pour chaque nouvelle arrivée d'un point de données. Cependant, il est presque toujours multivarié, un point de données étant décrit par plusieurs caractéristiques, ce qui augmente le risque de fausses alertes. Par ailleurs, Baier et al., 2021 indique également que ce type de méthode peut détecter des dérives sur les caractéristiques qui ne se répercutent pas sur les performances du modèle d'apprentissage automatique. Enfin, la surveillance de ce flux ne permettra de détecter que des dérives virtuelles.

— Le deuxième flux qu'il est possible de surveiller est celui des mesures d'incertitude associées à chaque prédiction, c'est-à-dire les $m(X_i, f)$. Ces mesures correspondant aux mesures d'utilité en apprentissage actif. Comme pour le flux des caractéristiques, une valeur est générée pour chaque point de données transmis par le flux. Ce flux est notamment utilisé pour la détection de dérives par Baier et al., 2021, qui utilisent les incertitudes associées aux prédictions de réseaux de neurones bayésiens. Sous l'hypothèse que cette mesure d'incertitude est corrélée à l'erreur de prédiction, surveiller ce flux permet de détecter des baisses de performance du modèle. Cependant, comme pour le flux de caractéristiques, surveiller ce flux d'utilité des prédictions ne permet de détecter que des dérives virtuelles.

— Le troisième flux considéré est celui des erreurs de prédiction. L'inconvénient principal d'utiliser ce flux est que ces erreurs ne sont disponibles que pour la fraction des points de données transmis pour laquelle des simulations sont lancées et le vrai label obtenu. De plus, ces vrais labels ne sont obtenus qu'après la fin des simulations, et non à la transmission des points de données. C'est, cependant, le seul flux permettant de détecter des dérives exclusivement réelles.

3.3 Implémentation et environnement expérimental

Un schéma illustrant l'intégration de la stratégie de couplage proposé à une ombre numérique est proposé figure 3.2. L'ombre numérique peut-être séparée en trois composants. Le premier est associé au traitement et au stockage des données. C'est en particulier ce composant que conserve l'historique des données traitées par l'ombre et les prédictions réalisées, pour l'aide à la décision et la mise à jour des modèles. Le deuxième composant rassemble les différents modèles intégrés à l'ombre numérique. On y retrouve, en particulier, le modèle de simulation, le modèle de substitution et potentiellement d'autres modèles d'aide à la décision ou la détection de dérive. Enfin, le troisième composant réalise la coordination entre les données et modèles intégrés à l'ombre numérique. C'est dans ce composant, en particulier, qu'est prise la décision d'utiliser ou non le modèle de simulation pour réaliser une prédiction.

Le fait que la prise de décision soit, ici, représentée dans l'espace réel et non dans l'espace virtuel est propre à l'utilisation envisagée de l'ombre numérique comme outils d'aide à la décision et pourrait être amené à changer suivant le cas d'usage.

Par ailleurs, l'implémentation objet de la stratégie proposée est ici basée sur quatre classes principales. Le premier est la classe *MLModel*, contenant le modèle d'apprentissage automatique. Le processus de structuration des données modélisé par la fonction h étant, ici, spécifique au modèle d'apprentissage automatique, cette fonction est également intégrée à cette classe. Le deuxième est un objet de la classe *SimulationModel* qui contient le modèle de simulation. A ces deux objets s'ajoute un objet *DriftDetector* qui surveille le flux choisi pour la détection de dérives conceptuelles et lance une alerte lorsqu'un tel évènement est détecté. Enfin, la classe *Router* coordonne les différents éléments de l'ombre numérique. La figure 3.3 présente le diagramme

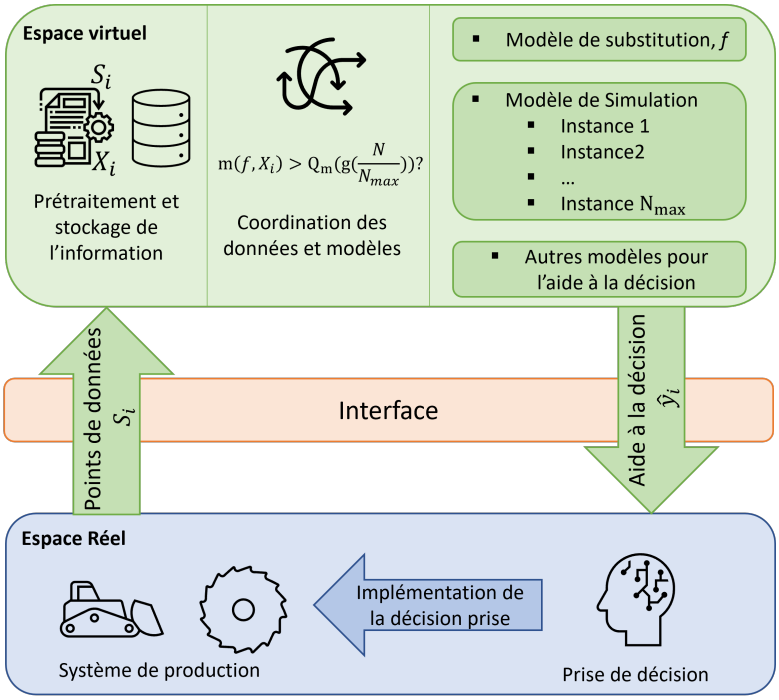

FIGURE 3.2 : Intégration de la stratégie de couplage proposée dans une ombre numérique

d'activités du processus lancé à la transmission d'un nouveau point de données par le flux contrôlé pour la détection de dérives. Plus précisément, le cas considéré est celui ou le flux surveillé est le flux des erreurs de prédictions pour les points de données pour lesquels une simulation est réalisée. En pratique, ces points de données sont représentés par des objets de la classe *DataItem*, complétés au fur et à mesure que de nouvelles informations sont générées par le modèle d'apprentissage ou le modèle de simulation. Ainsi, comme détaillé dans la section précédente, un point de données transmis par le flux d'arrivée est d'abord complété avec la prédiction du label faite par le modèle d'apprentissage, puis par la mesure d'utilité (ou d'incertitude) associée à cette prédiction. Cette mesure d'utilité est utilisée pour décider d'utiliser ou non le modèle de simulation pour obtenir le vrai label du point de données. S'il est décidé d'utiliser le modèle de simulation, le point de données peut être complété par son vrai label, puis être utilisé pour la détection de dérives. Il peut alors être décidé de réentrainer le modèle d'apprentissage automatique, si une dérive a été détectée, et si suffisamment de nouvelles données labélisées ont été obtenues, ou si toute autre condition spécifique au cas d'usage est réalisée.

En plus de ces quatre classes de base et de la classe DataItem représentant les points de données, d'autres classes ont été implémentées pour permettre l'expérimentation et l'évaluation de plusieurs variantes de la stratégie de couplage proposée à partir de jeux de données de résultats de simulations précalculés ou d'autres jeux de données de benchmark mis à disposition par la littérature.

Dans les expériences numériques réalisées dans la littérature traitant de l'apprentissage automatique en flux, les flux de données sont souvent générés en créant simplement un ordre d'arrivée aléatoire pour les points d'un jeu de données. Cette méthode ne considère cependant ni la variabilité des temps d'inter-arrivées, ni la variabilité des temps de simulation. Les classes ajoutées

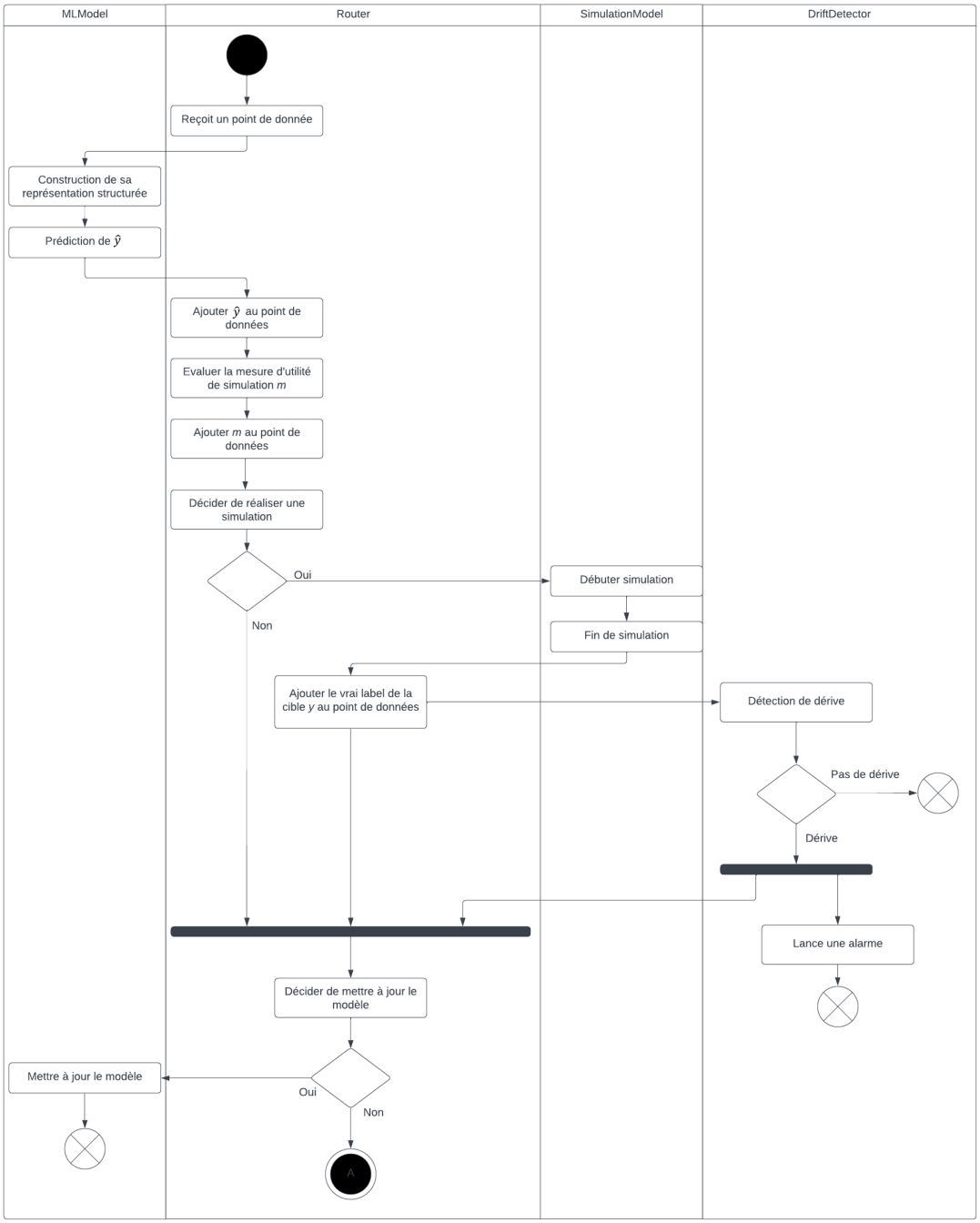

FIGURE 3.3 : Diagramme d'activité simplifié des évènements déclenchés par l'arrivée d'un point de données transmis par le flux.

implémentent donc, ici, un environnement expérimental basé sur la simulation à évènements discrets. La figure 3.4 présente un diagramme de classes simplifié des classes implémentées. Sont mises en bleu les classes correspondant à l'ombre numérique. On y retrouve notamment des objets *MLModel*, *SimulationModel*, *DriftDetector* et *Router* déjà présentés. Telle qu'implémentée durant ces travaux de thèse, la classe *SimulationModel* n'est utilisée que pour émuler le comportement des serveurs réalisant les simulations à partir de jeux de données précalculés. En particulier, elle contient une liste *SimulationInstance* contenant les temps restant des simulations en cours. L'objet *DataItemDB* représente une base de données utilisée pour stocker tous les points de données au fur et à mesure qu'ils sont transmis par le flux d'arrivée et complétés par les différents modèles. L'objet *Stream* n'est pas intégré à l'ombre numérique mais correspond au flux de données entrantes. Enfin, la classe *Interface* sert d'interface entre l'objet flux et l'ombre numérique et contrôle l'environnement de simulation à évènement discret constituant l'environnement expérimental.

3.4 Étude en régime stationnaire

L'objectif de cette section est de présenter des résultats théoriques préliminaires étudiant le comportement du système considéré. Nous nous concentrons sur le cas du régime stationnaire, en utilisant la théorie des files d'attente.

Les hypothèses et notations sont les suivantes :

- Le flux d'arrivée transmet de nouveaux points de données les uns à la suite des autres. Les temps d'inter-arrivées entre deux points de données sont des variables aléatoires indépendantes et identiquement distribuées (iid). Notons μ le temps moyen entre deux arrivées.
- Les temps de simulation nécessaires à l'obtention d'un label par le simulateur sont également des variables aléatoires positives iid. Notons λ le temps moyen de simulation.
- Le temps de simulation est supposé indépendant de l'erreur de prédiction commise par le modèle d'apprentissage automatique.
- Les mesures d'utilités $m(f, X_k)$ sont iid.
- $K = K(t)$ est le nombre d'arrivées jusqu'à l'instant t. N_k est le nombre de serveurs occupés durant la k^e arrivée. De même, X_k et y_k désignent le vecteur de caractéristiques et le label du k^e point de données. On pourra, en général, distinguer $\hat{y}_{ML,k}$, la prédiction du modèle d'apprentissage automatique et $\hat{y}_{Simul,k}$ la prédiction du modèle de simulation. Il est, cependant, supposé que le modèle de simulation est très fidèle, et que, en particulier, $\hat{y}_{Simul,k} = y_k$. Ainsi, si aucune précision supplémentaire n'est donnée, $\hat{y}_k$ sera utilisé pour les prédictions du modèle d'apprentissage automatique et y_k pour celles du modèle de simulation.
- l note la fonction de perte utilisée pour évaluer l'erreur commise pour une prédiction. l_k désignera l'erreur pour la k^e prédiction, $l_k = l(\hat{y}_k, y_k)$.

Taux d'échantillonnage du flux

Nous nous concentrons, dans un premier temps, sur le cas ou les temps de simulation et d'inter-arrivées suivent chacun une loi exponentielle. Dans ce cas, le processus définit par les $(N_t)_{t\in\mathbf{R}+}$ est un processus markovien de saut sur $[\![0, N_{max}]\!]$ de générateur infinitésimal :

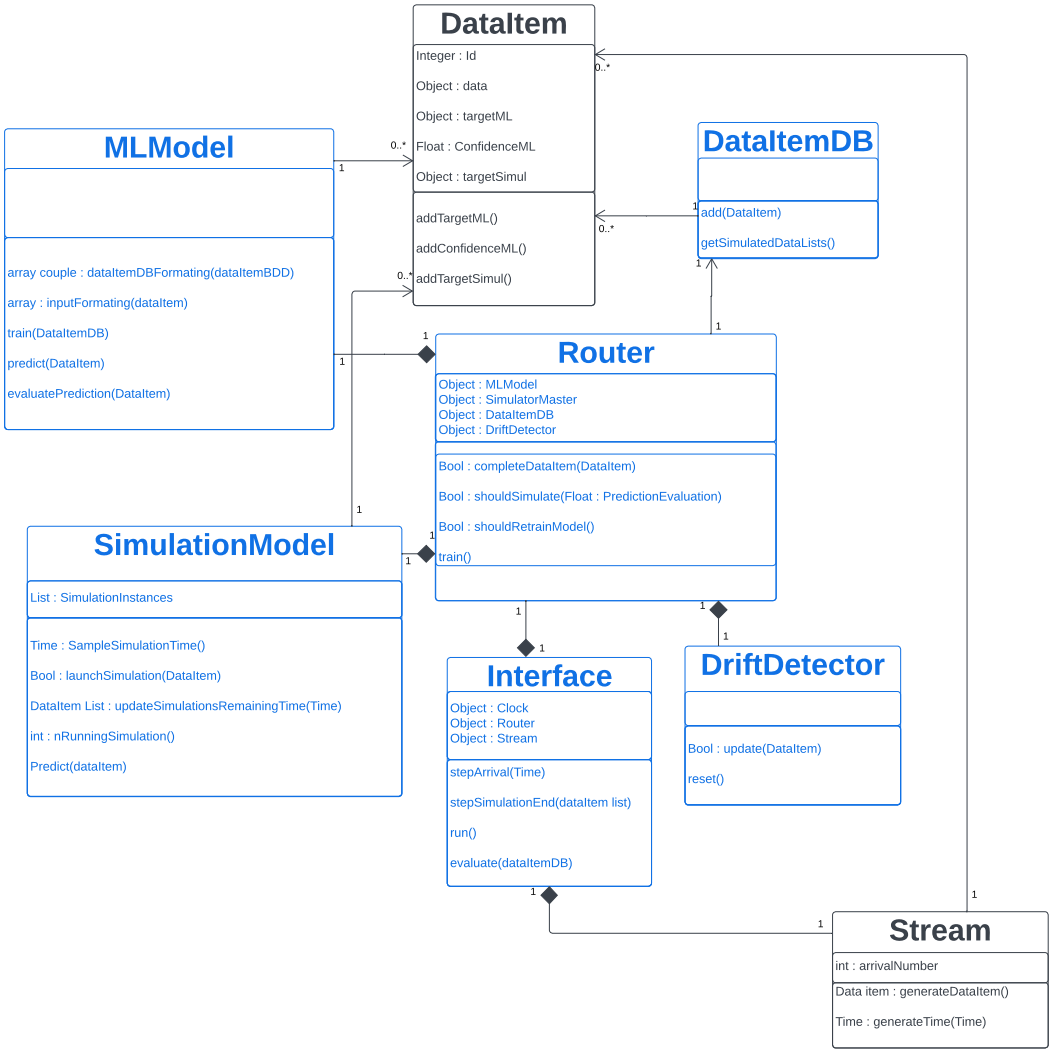

FIGURE 3.4 : Diagramme de classes simplifié de la stratégie proposée

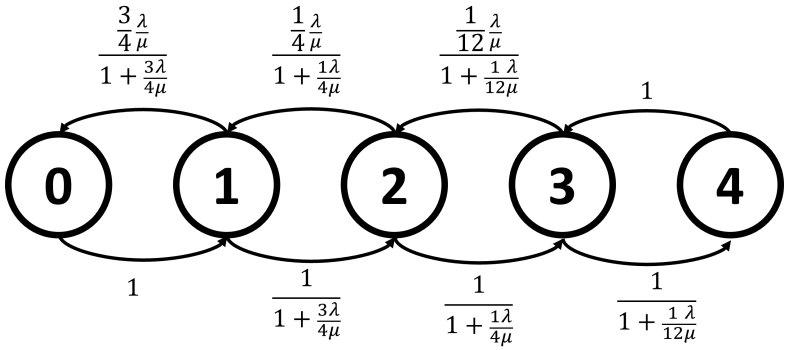

FIGURE 3.5 : Graphe de markov de la chaîne incluse dans le cas $N_{max} = 4$.

$$Q = \begin{bmatrix} -\frac{1}{\mu} & \frac{1}{\mu} & 0 & & \dots & & 0 \\ \frac{1}{\lambda} & -\frac{1}{\lambda} - \frac{N_{max}-1}{N_{max}\mu} & \frac{N_{max}-1}{N_{max}\mu} & 0 & \dots & & \\ 0 & \frac{2}{\lambda} & -\frac{2}{\lambda} - \frac{N_{max}-2}{N_{max}\mu} & \frac{N_{max}-2}{N_{max}\mu} & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \\ & & & & \dots & \frac{N_{max}}{\lambda} & -\frac{N_{max}}{\lambda} \end{bmatrix} \tag{3.4}$$

En effet, considérons le processus à un instant t quelconque tel que $N_t = n$. Le temps du prochain évènement est décidé par la première sonnerie d'un ensemble d'horloges exponentielles. Le processus transitionnera vers le niveau $n-1$ au prochain temps de saut si la première horloge à sonner est une des n horloges exponentielles de taux $\frac{1}{\lambda}$ correspondant à une fin de simulation. Ces n horloges sont équivalentes à une horloge exponentielle de taux $\frac{n}{\lambda}$. Ces termes correspondent à ceux de la sous-diagonale du générateur. De même, le processus transitionnera vers le niveau $n+1$ si la première horloge à sonner est une horloge correspondant à l'arrivée d'un point de données X_k tel que $m(f, X_k) \geq Q_m(\frac{n}{N_{max}})$. En particulier, $\mathbb{1}_{m(f,X_k)\geq Q_m(\frac{n}{N_{max}})}$ suit une loi de Bernouilli de paramètre $\frac{N_{max}-n}{N_{max}}$. Ces arrivées sont équivalentes à celles générées par un processus de poisson de paramètre [3] : $\frac{N_{max}-n}{N_{max}\mu}$. Ces termes correspondent à la sur-diagonale du générateur infinitésimal.

Le graphe de Markov associé à la chaîne de Markov incluse dans ce processus de saut est présenté figure 3.5 dans le cas ou $N_{max} = 4$. Ainsi, la probabilité pour la chaîne de sauter vers un état plus haut est d'autant plus faible que le nombre de serveurs occupés N est grand. De plus la chaîne est irréductible récurrente et positive. Notamment, il est possible de passer de n serveurs utilisés à 0 en n ou $n+1$ sauts avec une probabilité non nulle, si toutes les nouvelles arrivées sont rejetées ou toutes sauf une.

Par ailleurs, la résolution du système $\pi Q = 0$ a pour unique solution le vecteur $(\pi_0, \dots, \pi_N)$ avec :

3. `https://perso.math.univ-toulouse.fr/lagnoux/files/2013/12/master2_exos.pdf`, section 2.4.1

$$\pi_i = (\frac{\lambda}{N_{max}\mu})^i \binom{N_{max}}{i} \pi_0 \,, \tag{3.5}$$

avec :

$$\pi_0 = \frac{1}{(1+\frac{\lambda}{N_{max}\mu})^N} \tag{3.6}$$

La distribution de la chaîne tend vers π_0 et $\mathbf{E}_{\pi_0}(N) = \frac{N_{max}}{1+\frac{N_{max}\mu}{\lambda}}$. De plus, le théorème ergodique assurant la convergence des moyennes temporelles vers les espérances mathématiques s'applique.

Ainsi, le taux d'échantillonnage du flux est donné par :

$$\mathbf{P}(m(f,X) \geq Q_m(\frac{N}{N_{max}})) = \mathbf{E}(\mathbb{1}_{m(f,X)\geq Q_m(\frac{N}{N_{max}})}) = \frac{1}{1+\frac{\lambda}{N_{max}\mu}} \tag{3.7}$$

Dans le cas général, la loi de Little (Little et Graves, 2008) permet d'obtenir le même résultat. Le calcul repose cependant sur des hypothèses d'ergodicité et de stationnarité du processus à temps continu $(N_t)_{t\in\mathbf{R}+}$ d'une part, et du processus à temps discret $(N_k, S_k, m(f,X_k))$ d'autre part. En effet, la loi de Little (Little et Graves, 2008) donne la relation suivante entre la moyenne temporelle du nombre de serveurs occupés, $\langle N\rangle = \lim_{T\to\infty}\langle N\rangle_T = \lim_{T\to\infty}\frac{1}{T}\int_0^T N_t dt$ et le temps moyen de service W, définit par $W = \lim_{T\to\infty} W_T = \lim_{T\to\infty}\frac{\langle N\rangle_T}{K(T)}$:

$$\langle N\rangle = \frac{W}{\mu} \tag{3.8}$$

Le temps de service pour une arrivée est considéré égal à 0 si la prédiction est réalisée par le modèle d'apprentissage automatique, et de moyenne λ sinon. Ainsi, le temps de service moyen peut être écrit comme :

$$\begin{aligned}
W &= \lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}_{m(f,X_k)\geq Q_m(\frac{N}{N_{max}})}S_k && (3.9)\\
&= \mathbf{E}(\mathbb{1}_{m(f,X)\geq Q_m(\frac{N}{N_{max}})}S) && (3.10)\\
&= \mathbf{E}(\mathbb{1}_{m(f,X)\geq Q_m(\frac{N}{N_{max}})})\mathbf{E}(S) && (3.11)\\
&= (1-\frac{\mathbf{E}(N)}{N_{max}})\lambda && (3.12)\\
&= (1-\frac{\langle N\rangle}{N_{max}})\lambda && (3.13)
\end{aligned}$$

Ainsi,

$$\langle N \rangle = (1 - \frac{\langle N \rangle}{N_{max}})\frac{\lambda}{\mu} \tag{3.14}$$

Ce qui permet encore d'obtenir le taux d'échantillonnage du flux :

$$\mathbf{P}(m(f,X) \geq Q_m(\frac{N}{N_{max}})) = \mathbf{E}(\mathbb{1}_{m(f,X) \geq Q_m(\frac{N}{N_{max}})}) = \frac{1}{1 + \frac{\lambda}{N_{max}\mu}} \tag{3.15}$$

Erreur préquentielle du couple

Une stratégie de couplage entre le modèle de simulation et son métamodèle utilisant la méthode décrite par l'équation 3.3 peut-être évaluée à partir de l'erreur préquentielle du couple :

$$EPC_K = \frac{1}{K}\sum_{k=1}^{K} l(\hat{y}_{ML,k}, y_k)\mathbb{1}_{m(f,X_k) \leq \frac{N_k}{N_{max}}} + \frac{1}{K}\sum_{k=1}^{K} l(\hat{y}_{Simul,k}, y_k)(1 - \mathbb{1}_{m(f,X_k) \leq \frac{N_k}{N_{max}}}) \tag{3.16}$$

$$= \frac{1}{K}\sum_{k=1}^{K} l(\hat{y}_k, y_k)\mathbb{1}_{m(f,X_k) \leq \frac{N_k}{N_{max}}} \tag{3.17}$$

En faisant, encore une fois, les hypothèses de stationnarité et d'ergodicité du processus $(N_k, m(f,X_k), l(\hat{y}_k, y_k))$, cette erreur préquentielle converge vers $\mathbf{E}(l_k \mathbb{1}_{m(f,X) \leq \frac{N}{N_{max}}})$. Il est également informatif d'écrire cette limite en fonction de la corrélation ρ entre $\mathbf{E}(l_k$ et $\mathbb{1}_{m(f,X) \leq \frac{N}{N_{max}}})$:

$$\mathbf{E}(l\mathbb{1}_{m(f,X) \leq \frac{N}{N_{max}}}) = (1-b)\mathbf{E}[l] - \rho\sqrt{b(1-b)\mathbf{Var}[l]} \tag{3.18}$$

avec $b = \mathbf{P}(m(f,X) \geq Q_m(\frac{N}{N_{max}}))$ le taux d'échantillonnage du flux. En particulier, l'erreur préquentielle sera d'autant plus faible que la corrélation entre $\mathbb{1}_{m(f,X) \leq \frac{N}{N_{max}}}$ et $l(\hat{y}, y)$ est grande.

Il est alors possible d'inférer une condition pour que cette erreur asymptotique soit plus faible que celle d'une stratégie échantillonnant le flux au taux maximal $b_{max} = N_{slots}\frac{\mu}{\lambda}$ indépendamment de la mesure $m(f,X)$. Ce taux maximal est obtenu en appliquant la loi de Little et en remarquant que $\mathbf{E}(N) = \frac{\mathbf{E}(\mathbb{1}_{\mathbf{ToSimul}}\mathbf{S})}{\mu} = b\frac{\lambda}{\mu} \leq N_{slots}$.

En particulier, comparer l'erreur préquentielle obtenue avec cette stratégie avec celle obtenue par l'équation 3.18 donne :

$$\rho > \frac{\mathbf{E}[l]}{\sqrt{\mathbf{Var}[l]}} \times \frac{b_{max} - b}{\sqrt{b(1-b)}} \tag{3.19}$$

$$\rho > \frac{\mathbf{E}[l(\hat{y}_{ML,k})]}{\sqrt{\mathbf{Var}[l(\hat{y}_{ML,k}, y_k)]}} \times (N_{slot}\frac{\mu}{\lambda})^{\frac{3}{2}} \tag{3.20}$$

Ainsi, la stratégie d'échantillonnage basée sur les mesures d'utilité des prédictions aura d'autant plus de chances de mener à une erreur préquentielle du couple inférieure à celle observée pour une stratégie maximisant le taux d'utilisation des serveurs que l'espérance des erreurs individuelles est petite devant leur variance et que le temps moyen de simulation est grand devant le temps moyen d'inter-arrivées.

3.5 Premières évaluations

Cette section présente les premiers résultats expérimentaux. L'objectif est de comparer plusieurs variantes de la stratégie de couplage proposée sur huit jeux de données de benchmark utilisés dans la littérature scientifique. Ces jeux de données ne sont pas composés de résultats de simulations, mais présentent l'avantage de proposer des tâches de prédiction provenant de domaines d'application très variés. Le modèle d'apprentissage automatique utilisé pour ces expériences est un modèle de forêt aléatoire. Ce modèle a été sélectionné pour ses bons résultats dans de nombreuses applications à des problèmes industriels, incluant l'application présentée dans les chapitres 4 et 5 de ce mémoire.

3.5.1 Jeux de données UCI

Les expériences présentées dans cette section sont menées sur huit jeux de données provenant du dépôt UCI [4]. Ces huit jeux de données ont été sélectionnés car les tâches d'apprentissage associées sont des tâches de régressions, et car ils contiennent chacun plus de 1000 points de données. Les caractéristiques principales de ces jeux de données sont présentées table 3.1.

La tâche associée au jeux de données Wine Quality (Cortez et al., 2009) est la prédiction d'un indicateur de qualité d'un vin à partir de ses propriétés physico-chimiques. Celle associée au jeux de donnée Appliance energy prediction (Candanedo, Feldheim et Deramaix, 2017) est la prédiction de la consommation électrique d'un ensemble d'appareils électroménagers à partir de données de température et d'humidité relevées dans diverses pièces d'un bâtiment, ainsi que de données météorologiques. L'objectif du troisième jeu de données, Combine Cycle Power Plant (Tüfekci, 2014), est la prédiction heure par heure de la production d'une centrale électrique à partir de données météorologiques et physiques prises sur la turbine. La tâche associée au quatrième jeu de données, Gaz turbine Co and NOx emissions (Kaya, Tüfekci et Uzun, 2019) est de prédire le niveau d'émission par une turbine de deux types de gaz à partir de données météorologiques et de relevés pris sur la turbine en fonctionnement. L'objectif du jeu de données Electrical grid stability (Arzamasov, Böhm et Jochem, 2018) est de prédire un indicateur de stabilité d'un réseau électrique à partir du comportement de fournisseurs et de consommateurs. Il s'agit d'un jeu de données simulé. La tâche de prédiction associée au jeu de données Concrete compressive strenght (Yeh, 1998) est de prédire la résistance en compression de bétons à partir de leur âge et composition. Le jeu de données Geographical origin of music (Zhou, Claire et King, 2014) a pour objectif la prédiction de la latitude et de la longitude de la capitale du pays d'origine de morceaux de musique à partir de caractéristiques extraites des bandes sonores. Enfin, le jeu de données Superconductivity data (Hamidieh, 2018) a pour objectif la prédiction de la

4. `https://archive.ics.uci.edu/ml/index.php`

Nom	**Référence**	**Nombre de points de données**	**Nombre de caractéristiques**	**Dimension de la cible**
Wine quality (WQ)	Cortez et al., 2009	4898	11	1
Appliance energy prediction (AE)	Candanedo, Feldheim et Deramaix, 2017	19735	27	1
Combined cycle power plant (CCPP)	Tüfekci, 2014	9568	4	1
Gaz turbine CO and NOx emissions (GT)	Kaya, Tüfekci et Uzun, 2019	7411	10	2
Electrical grid stability (EG)	Arzamasov, Böhm et Jochem, 2018	10000	12	1
Concrete compressive strength (CCS)	Yeh, 1998	1030	8	1
Geographical origin of music (GOM)	Zhou, Claire et King, 2014	1059	68	2
Superconductivity data (SD)	Hamidieh, 2018	21263	81	1

Table 3.1 : Caractéristiques principales des jeux de données UCI.

température critique de superconducteurs à partir de caractéristiques extraites de leurs formules chimiques.

Ces jeux de données ne sont pas, pour la plupart, des jeux de données de résultats de simulation. Ils présentent cependant l'intérêt d'être librement accessibles, déjà utilisés dans la littérature scientifique, et de provenir de domaines d'application très variés. Dans les expériences qui suivent, ces jeux de données sont utilisés pour générer des flux de données, qui sont, dans ce contexte, constitués d'un ordre aléatoire d'un jeu de données couplé à des temps d'inter-arrivées eux aussi aléatoires. Pour pouvoir analyser les résultats des expériences proposées, il est nécessaire de contrôler quelles sections des flux sont stationnaires, ainsi que les périodes de dérive. Certains des jeux de données présents, sont, cependant, déjà des séries temporelles, ce qui induit le risque que des dérives soit déjà présentes. A ce titre, les caractéristiques correspondant à des dates ont systématiquement été retirées et les jeux de données réordonnés aléatoirement.

3.5.2 Comparaison de plusieurs mesures d'utilités

Cette section compare plusieurs méthodes d'échantillonnage du flux de données d'entrée, c'est-à-dire de plusieurs stratégies choisissant quels points de données labéliser ou non à l'aide du modèle de simulation.

Les quatre premières méthodes utilisent des mesures de l'incertitude associées à chaque prédiction qui sont comparées au seuil adaptatif définit par l'équation 3.3. Ces mesures d'incertitudes sont tirées ou inspirées de la littérature traitant de l'apprentissage actif. Deux de ces mesures sont définies uniquement pour des prédictions unidimensionnelles, c'est-à-dire à valeur dans $\mathbf{R}$. Lorsque le label est un vecteur de dimension J, $J > 1$, c'est-à-dire lorsque le label prédit est à valeur dans $\mathbf{R}^J$, la mesure est adaptée tel que $m(X) = \sum_{j=1}^{J} m_j(X)$, avec $m_j(X)$ la mesure initiale appliquée à la j^e dimension du label, indépendamment des autres.

La première mesure considérée est nommée *ambiguité* dans Burbidge, Rowland et King, 2007. Elle correspond à la variance des prédictions des différents arbres de la forêt :

$$a(X) = \frac{1}{B-1} \sum_{i=1}^{B} (\hat{y}_i(x) - \hat{y}(x))^2 \,. \tag{3.21}$$

La prédiction de la forêt est en effet donnée par $\hat{y}(X) = \frac{1}{B} \sum_{i=1} \hat{y}_{.i}(X)$.

La seconde mesure considérée est un estimateur de la *variance d'échantillonnage* de la prédiction, cette mesure évalue le degré de changement de la prédiction si la forêt avait été entraînée sur une base d'entraînement différente. Cet estimateur a été développé par Wager, Hastie et Efron, 2014 et est utilisé pour l'apprentissage actif d'un modèle de forêt aléatoire dans Farache et al., 2022. Il est défini comme :

$$s(x) = \sum_{\omega=1}^{\Omega} Cov[J_{i\omega}, \hat{y}_{.}i(x)]^2 + \sum_{\omega=1}^{\Omega} (\hat{y}_{-\omega}(x) - \hat{y}(x))^2 - \frac{e\Omega}{B} a(x) \tag{3.22}$$

Ω est le nombre d'exemples d'entraînement. $J_{i\omega}$ compte le nombre de fois que le ω^e exemple est

présent dans la base d'entraînement du i^e arbre de la forêt aléatoire. *Cov* est l'estimateur usuel de covariance. $\hat{y}_{-\omega}(x)$ est la prédiction faite par la forêt après avoir retiré tous les arbres entraînés en utilisant le ω^e exemple.

La troisième mesure, nommée *consistance*, a été introduite par Gao et al., 2020 dans le cadre de l'apprentissage actif pour la classification d'images. Cette mesure est motivée par l'idée qu'une prédiction ne devrait pas être sensible à des transformations des caractéristiques décrivant le point de données ne changeant pas le label. Pour la classification d'images, ces transformations peuvent, par exemple, être des rotations ou des symétries. Plus généralement, les prédictions devraient être peu sensibles à l'application de bruit sur les données d'entrée. La *consistance* est donc définie comme :

$$c(x) = Var(\hat{y}(\tilde{x}))\,, \tag{3.23}$$

avec $\hat{y}(\tilde{x})$ la prédiction de la forêt aléatoire pour une entrée $\tilde{x}$ générée en ajoutant un bruit blanc suivant une loi normale $N(0, \sigma_0)$ à x. Un inconvénient de cette méthode est la difficulté à fixer σ_0 a priori. Dans cette étude, il a été fixé par essais et erreurs à $\frac{\sigma}{10}$ avec σ la déviation standard de la caractéristique auquel il est ajouté.

La dernière mesure est la distance minimale, ou plus généralement la *dissimilarité* minimale, entre le point de données x et les données utilisées pour l'entraînement du modèle d'apprentissage :

$$diss(X) = \min_{x_i in D} d(X, x_i)\,, \tag{3.24}$$

avec D la base de données d'entraînement. Cette mesure est indépendante de la prédiction du modèle d'apprentissage utilisé. Elle est motivée par l'algorithme d'apprentissage actif Kernel Farthest First proposé dans Baram, Yaniv et Luz, 2004. Elle est également utilisée par Xu, Akella et Zhang, 2007 en conjonction avec deux autres mesures pour la classification de documents.

Deux autres stratégies sont utilisées pour référence. La première est la *maximisation naive* de l'usage de l'échantillonnage du flux, en décidant de lancer une simulation chaque fois qu'un point de données est transmis par le flux et que le nombre de serveurs occupés est inférieur à N_{max}. La deuxième stratégie donnée pour référence est particulière en ce sens que les expériences qui l'évaluent ne considèrent ni les temps d'inter-arrivées, ni les temps de simulation aléatoires. Un point de données est labélisé toutes les ν arrivées, ν un entier fixé choisi pour que le taux d'échantillonnage du flux avec cette stratégie soit identique à celui des stratégies utilisant les mesures d'apprentissage actif.

Pour isoler et comparer la capacité des différentes stratégies à minimiser l'erreur préquentielle du couple en sélectionnant les points de données pour lesquels l'erreur de prédiction est la plus importante, les modèles d'apprentissage utilisés sont systématiquement entraînés sur uniquement les 500 premiers points de données des flux, et jamais réentrainés sur les données nouvellement sélectionnés.

Dans les expériences présentées dans ce chapitre, le nombre de serveurs est fixé à 10, c'est-à-dire que jusqu'à 10 simulations peuvent tourner en parallèle. Les temps d'inter-arrivées aléatoires sont

générés suivant une loi exponentielle de moyenne $\mu = 1$. De même, les temps de simulation sont générés suivant une loi du χ_2 de paramètre λ. Le choix d'une loi du χ_2 est fait car cette loi génère des valeurs positives. De plus, pour des valeurs du paramètre λ assez grandes, elle est similaire à une loi normale. Enfin, elle ne dépend que d'un seul paramètre, correspondant à leur espérance. Nous constaterons également que, bien qu'elle soit très différente de la loi exponentielle, la formule du taux d'échantillonnage définie équation 3.7 est vérifiée. En particulier, il est ici d'environ 20% pour les stratégies basées sur l'apprentissage actif. L'impact du choix de ces paramètres sera étudié dans le chapitre 5.

Pour chaque jeu de données et stratégie d'échantillonnage, 30 flux ont été générés aléatoirement et utilisés pour évaluer les performances de la stratégie. L'objectif de ces 30 répétitions est de moyenner les impacts des variables aléatoires, nombreuses dans ces expériences. Elles incluent, en effet, tous les temps d'inter-arrivées, temps de simulation et ordre des flux. A la fin de chaque flux, trois quantités sont systématiquement relevées. La première est le taux d'échantillonnage du flux, c'est-à-dire la proportion des données d'entrée pour lesquelles une simulation a été lancée. La deuxième est l'erreur préquentielle du couple définie par l'équation 3.16. La fonction de perte utilisée est, ici, l'erreur quadratique $l(\hat{y}, y) = ||\hat{y} - y||^2$. Enfin, la troisième est l'erreur préquentielle des prédictions du modèle d'apprentissage automatique pour les points de données non simulés. Cette erreur est donc moins impactée que l'erreur préquentielle du couple par le taux d'échantillonnage du flux et évalue la capacité de la stratégie d'échantillonnage à sélectionner les points de données ayant l'erreur de prédiction la plus importante.

Les taux d'échantillonnage moyens de chaque stratégie sur chaque jeu de données sont présentés table 3.2. Les taux d'échantillonnage des quatre stratégies basées sur l'échantillonnage actif ont, en particulier, un taux d'échantillonnage proche de 20%. Le taux d'échantillonnage de la méthode maximisant naïvement l'usage du modèle de simulation est supérieur approximativement de 4% à celui des méthodes basées sur l'apprentissage actif, et très légèrement inférieur au seuil maximal de 25%

La figure 3.6 présente l'évolution des erreurs préquentielles des prédictions des points de données non simulés (EPPNS) pour chaque stratégie sur chacun des huit jeux de données UCI. Chaque sous graphique correspond à un jeu de donnée. Sur chaque sous graphique, chaque courbe correspond à une stratégie particulière évaluée sur le jeu de données correspondant. Ces courbes sont les courbes moyennes sur les 30 répétitions de chaque expérience. De plus, les intervalles bleus et orange contiennent en tout point 80% des courbes correspondant aux stratégies *maximisation naïve* et *ambiguïté* respectivement. Seul les intervalles correspondant à ces deux stratégies sont indiqués pour simplifier la lecture du graphique. La stratégie *ambiguïté* a, en particulier, été sélectionnée comme étant, dans l'ensemble, la stratégie la plus performante. Le code permettant l'obtention de ces courbes a été obtenu de la librairie Cardinal Abraham et Dreyfus-Schmidt, 2022.

Les flux de données générés étant ici stationnaires, les EPPNS convergent rapidement vers une valeur limite. Les stratégies *maximisation naïve* et *1 in ν* ont des comportements très semblables du point de vue des EPPNS, puisqu'aucune des deux ne considère les caractéristiques des points de données ou des prédictions dans la décision de faire ou non appel au modèle de simulation. Dans presque tous les cas, les EPPNS des stratégies basées sur les mesures d'apprentissage actif tendent vers des limites plus faible que celle des stratégies *maximisation naïve* et *1 in ν*. La seule exception est la stratégie basée sur la mesure de dissimilarité définie par l'équation 3.24 pour le jeu de données *Superconductivity data*. Les deux stratégies basées sur les mesures *ambiguité* et *variance d'échantillonnage* présentent des résultats très similaires. Ces deux mesures sont en effet

Jeu de données	**1 in ν**	**Maximisation naïve**	**Ambiguïté**	**Variance d'échantillonnage**	**Consistance**	**Dissimilarité**
WQ	0.200 (0.0)	0.241 (3.3×10^{-3})	0.200 (2.2×10^{-3})	0.200 (2.3×10^{-3})	0.200 (2.4×10^{-3})	0.200 (2.3×10^{-3})
AE	0.200 (0.0)	0.242 (1.8×10^{-3})	0.201 (1.3×10^{-3})	0.201 (1.3×10^{-3})	0.200 (1.3×10^{-3})	0.201 (1.3×10^{-3})
CCPP	0.200 (0.0)	0.241 (2.5×10^{-3})	0.200 (1.7×10^{-3})	0.200 (1.6×10^{-3})	0.200 (1.7×10^{-3})	0.200 (1.6×10^{-3})
GT	0.200 (0.0)	0.242 (3.6×10^{-3})	0.200 (2.4×10^{-3})	0.200 (2.5×10^{-3})	0.200 (2.6×10^{-3})	0.200 (2.6×10^{-3})
EG	0.200 (0.0)	0.242 (2.4×10^{-3})	0.201 (1.7×10^{-3})	0.200 (1.7×10^{-3})	0.200 (1.7×10^{-3})	0.200 (1.7×10^{-3})
CCS	0.200 (0.0)	0.233 (1.1×10^{-2})	0.195 (8.3×10^{-3})	0.195 (7.9×10^{-3})	0.195 (8.4×10^{-3})	0.195 (8.8×10^{-3})
GOM	0.199 (0.0)	0.231 (1.0×10^{-2})	0.194 (7.7×10^{-3})	0.193 (7.8×10^{-3})	0.193 (8.3×10^{-3})	0.193 (75×10^{-3})
SD	0.200 (0.0)	0.242 (1.7×10^{-3})	0.201 (1.4×10^{-3})	0.201 (1.4×10^{-3})	0.201 (1.3×10^{-3})	0.201 (1.2×10^{-3})

TABLE 3.2 : Taux d'échantillonnage moyen des 6 stratégies comparées sur les huit jeux de données UCI. Les déviations standards entre les résultats des 30 répétitions de chaque expérience sont indiquées entre parenthèses.

basées sur la même idée d'estimer des intervalles de confiance autour des prédictions des forêts aléatoire. Ces deux stratégies sont également celles présentant les EPPNS les plus faibles.

Les erreurs préquentielles du couple (EPC) sont présentées table 3.3. Les EPC les plus faibles sont systématiquement obtenues pour les stratégies utilisant les mesures *ambiguïté* ou *variance d'échantillonnage*, et ce malgré les taux d'échantillonnage de ces stratégies plus faibles que celui de la stratégie *maximisation naïve*. De plus, ces deux stratégies présentent toujours des EPC plus faibles ou égales à celles de la stratégie *maximisation naïve*. Le plus faible nombre de labels exacts obtenus auprès du simulateur est donc compensé par une meilleure sélection des données labélisées. Ce n'est, cependant, pas le cas pour les deux autres stratégies basées sur les mesures *consistance* et *diversité*. La stratégie basée sur la mesure *consistance* a une EPC plus haute que la stratégie *maximisation naïve* pour deux des huit jeux de données. De même, la stratégie basée sur la mesure *diversité* a une EPC plus haute sur trois des huit jeux de données.

Pour poursuivre l'analyse de ces résultats, les six stratégies comparées ont été systématiquement ordonnées par EPC moyenne croissante sur chacun des huit jeux de données UCI. Les rangs moyens de chaque stratégie sur les huit jeux de données UCI sont présentés dans le diagramme critique figure 3.7. Les stratégies basées sur les mesures *ambiguïté* et *variance d'échantillonnage* ont, sans surprise, les rangs les plus élevés. Les stratégies basées sur les mesures *consistance* et *dissimilarité* ont, cependant, des rangs moyens similaires à la stratégie *maximisation naïve*. Un test statistique de Friedman a été réalisé pour tester l'hypothèse nulle selon laquelle toutes les stratégies ont le même rang moyen. La p-value de ce test est 7.4×10^{-5}. L'hypothèse nulle est donc rejetée et il est possible d'affirmer que certaines stratégies ont, en moyenne, un rang plus élevé sur l'ensemble des jeux de données. De plus, des tests de Wilcoxon ont été réalisés pour comparer par paires les rangs médians des différentes stratégies. Les résultats de ces tests apparaissent sur le diagramme critique sous la forme de barres épaisses reliant plusieurs stratégies. Ces barres relient des groupes de stratégies pour lesquels les tests par paires rejettent systématiquement l'hypothèse nulle au niveau 5%. En particulier, les expériences réalisées ne permettent pas de conclure à une différence de rang significative entre les stratégies basées sur les mesures *ambiguité* et *variance d'échantillonnage*. De même pour les stratégies basées sur les mesures *diversité*, *consistance* et la stratégie *maximisation naïve*.

3.5.3 Comparaison de méthodes de détection de dérive

Cette section présente une étude comparative de plusieurs méthodes de détection de dérives. Plus particulièrement, le flux surveillé pour la détection de dérives est ici, le flux des erreurs de prédiction pour les points de données dont le vrai label a été obtenu par simulation. L'avantage principal de surveiller ce flux est, en général, de pouvoir détecter tout type de dérive, y compris des dérives purement réelles. Ce type de dérive n'est pas attendu dans le cas spécifique étudié au sein de cette thèse. Il peut être espéré qu'un changement du modèle de simulation entraînant une dérive réelle soit notifié automatiquement au reste du système déclenchant la mise à jour du modèle d'apprentissage. Nous nous plaçons cependant ici dans le cadre plus général de l'apprentissage actif en flux. L'objectif est alors de mettre en lumière un problème pouvant être causé par le biais d'échantillonnage pour la détection de dérive. Le biais d'échantillonnage est un problème par ailleurs bien connus de l'apprentissage actif (Dasgupta, 2011). Les données échantillonnées par apprentissage actif ne sont pas représentatives de la vrai distribution des données transmises par le flux. Cela peut causer plusieurs problèmes, allant de certaines stratégies d'échantillonnage sous échantillonnant des parties importantes de l'espace des caractéristiques et donc menant à

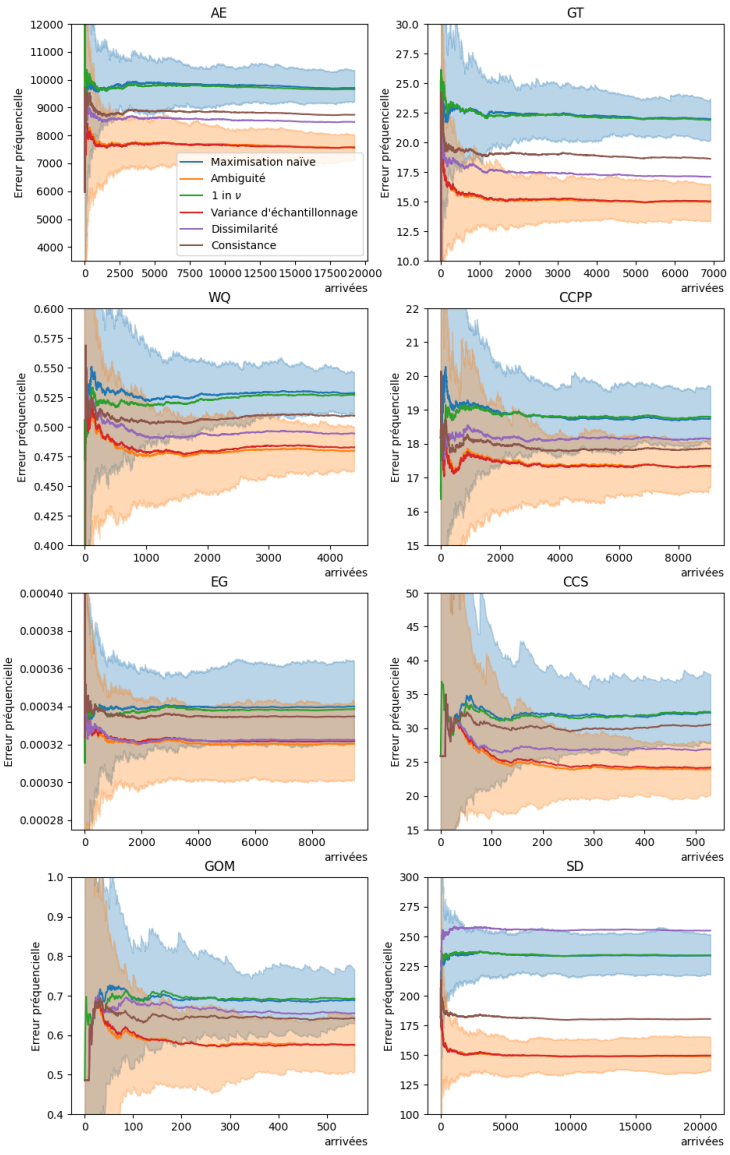

FIGURE 3.6 : Évolution de l'erreur préquentielle des points de données non simulés (EPPNS), indexés par ordre d'arrivée. Chaque courbe correspond à une stratégie d'échantillonnage. Ces courbes présentent la moyenne de l'erreur préquentielle sur 30 répétitions de l'expérience. Les intervalles bleus et orange contiennent en tout point 80% des courbes pour les stratégies *maximisation naïve* et *ambiguïté*.

Jeux de données	**1 in ν**	**Maximisation naïve**	**Ambiguité**	**Variance d'échantillonnage**	**Consistance**	**Diversité**
WQ	0.422 (0.012)	0.400 (0.011)	**0.383 (0.012)**	0.385 (0.011)	0.407 (0.012)	0.395 (0.010)
AE	7727 (363)	7353 (351)	6057 (353)	**6047 (266)**	6985 (370)	6779 (255)
CCPP	15.0 (0.496)	14.2 (0.473)	**13.9 (0.463)**	**13.9 (0.462)**	14.3 (0.412)	14.5 (0.418)
GT	17.5 (1.07)	16.6 (1.03)	**12.0 (0.97)**	**12.0 (0.97)**	14.9 (1.25)	13.5 (0.65)
EG	e 2.71×10^{-4} (1.47×10^{-5})	2.57×10^{-4} (1.42×10^{-5})	$\mathbf{2.56 \times 10^{-4}}$ ($\mathbf{1.44 \times 10^{-5}}$)	2.57×10^{-4} (1.43×10^{-5})	2.67×10^{-4} (1.46×10^{-5})	2.58×10^{-4} (1.39×10^{-5})
CCS	26.0 (2.76)	24.2 (2.98)	**19.0 (2.52)**	19.2 (2.42)	24.2 (2.92)	21.3 (2.91)
GOM	0.555 (0.041)	0.518 (0.048)	**0.457 (0.041)**	**0.457 (0.039)**	0.510 (0.043)	0.521 (0.038)
SD	187.2 (12.88)	177.2 (12.19)	**119.0 (8.30)**	119.5 (8.48)	144.1 (14.53)	203.7 (12.89)

TABLE 3.3 : EPC des différentes stratégies d'échantillonnages comparées sur les huit jeux de données UCI. les valeurs sont moyennées sur 30 répétitions de l'expérience. les déviations standards sont indiquées entre parenthèses. les erreurs les plus basses sont affichées en gras.

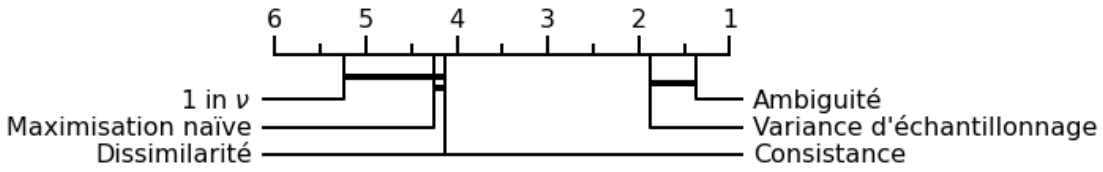

FIGURE 3.7 : Diagramme critique des différentes mesures d'utilité sur les huit datasets UCI. Le code permettant l'obtention de ce graphique est adapté de ISMAIL FAWAZ et al., 2019

des classifieurs sous-optimaux lorsque la base de données labélisées grandit, à rendre difficile l'évaluation et la comparaison non biaisée des modèles entraînés par apprentissage actif pour la sélection de modèle. Dans le cadre considéré ici, les données échantillonnées pour labélisation ont, avant l'apparition de dérives, une erreur moyenne plus grande que si elles étaient évaluées sur des données sélectionnées au hasard, ce qui peut être problématique pour des méthodes de détection de dérives cherchant à identifier des augmentations de la moyenne de cette erreur. Les méthodes basées sur des tests statistiques tels que Kolmogorov-Smirnov ou Anderson-Darling, cependant, considèrent la distribution d'erreurs dans son ensemble et la façon dont le biais d'échantillonnage peut influer sur les performances de ces méthodes n'est pas évidente.

Six des huit jeux de données UCI présentés table 3.1 sont utilisés dans les expériences présentées ici : Wine quality, Appliance energy prediction, Combined cycle power plant, Gaz turbine CO and NOx emissions, Electrical grid stability et Superconductivity data. Les jeux de données Concrete compressive strength et Geographical origin of music sont trop courts pour être utilisés ici, une fois retirées les données servant à l'initialisation des modèles. Pour chaque flux généré à l'aide d'un de ces jeux de données, une dérive est introduite après la moitié des arrivées en permutant aléatoirement les labels des différents point de données. L'objectif est de créer une dérive abrupte purement réelle, qui n'impacte en aucun cas le processus d'échantillonnage du flux. En effet, le fait d'échantillonner le flux en utilisant une mesure d'utilité ou d'incertitude peut amener à légèrement sur échantillonner le flux en cas de dérive virtuelle, ce qui pourrait accélérer la détection de dérive. Puisque dans ces expériences les modèles d'apprentissage ne sont pas re-entraînés après l'instant de dérive, il n'est pas nécessaire de conserver un lien logique entre les caractéristiques d'entrée et le label.

Les quatre méthodes de détection de dérives présentées table 2.1 sont comparées ici. L'implémentation incrémentale du test de Kolmogorov-Smirnov provient de REIS et al., 2016. Deux seuils ρ sur le test statistique effectué de manière répétée pour comparer une fenêtre glissante à une distribution de référence sont considérés : 0.005 et 0.001. Une implémentation incrémentale du test d'Anderson-Darling a été proposée et est présentée en annexe A. Deux seuils ρ sur le test statistique sont considérés : 0.005 et 0.001 qui sont les deux seuils les plus faibles tabulés dans SCHOLZ et STEPHENS, 1987. Considérant la diversité des tâches de prédiction considérées, les deux paramètres δ et h sont exprimés en multiples de l'écart-type σ du flux d'erreur surveillé. En particulier, δ est laissé fixe à $\delta = \frac{\sigma}{2}$ tandis que deux niveaux sont considérés pour le seuil h : 6σ et 10σ.

Pour mettre en lumière la problématique du biais d'échantillonnage, deux stratégies de sélection sont utilisées. La première est basée sur la mesure *ambiguïté*, qui montre le plus haut rang moyen dans le diagramme critique figure 3.7. La deuxième utilise le même mécanisme de sélection, mais

avec une "mesure d'utilité" générée aléatoirement, indépendamment de toute caractéristique du point de données ou de la prédiction. Les temps moyens d'inter-arrivées et de simulation ainsi que le nombre de serveurs est laissés tel que dans la section précédente.

Pour chaque stratégie de détection de dérives et chaque jeu de données, 100 flux sont générés aléatoirement. Chaque flux contient une dérive à la moitié des arrivées. Pour chaque flux, la première alerte lancée après la dérive est considérée comme son instant de détection. Toutes les autres alertes lancées sont considérées comme des fausses alertes. Il est possible, pour certains des flux les plus courts, qu'aucune alerte ne soit lancée après l'instant de dérive. Trois quantités sont évaluées sur chaque flux. La première est le taux de détection, c'est-à-dire la fraction des 100 répétitions pour lesquelles une dérive a été détectée. La deuxième est le nombre de fausses alarmes. Enfin, la troisième est le temps entre l'instant de dérive et l'instant d'arrivée du point de données lançant la première alarme après l'instant de dérive.

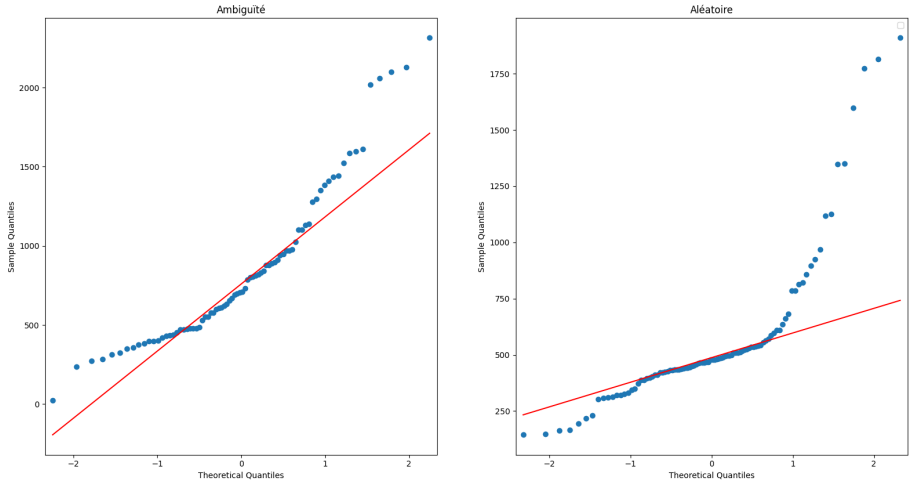

FIGURE 3.8 : Diagrammes quantiles-quantiles des temps de détection de dérives obtenues pour le jeu de données Wine Quality et la méthode de détection de dérives basée sur Kolmogorov-Smirnov avec $\rho = 0.005$. Le diagramme de gauche correspond au cas ou le flux d'arrivée est échantillonné avec la mesure d'incertitude *Ambiguïté*. Le diagramme de droite correspond au cas ou le flux d'arrivée est échantillonné avec une "mesure" d'incertitude aléatoire. La loi de référence est la loi normale.

Les résultats exacts de ces trois indicateurs calculés à partir des résultats obtenus sur les six jeux de données étudiés sont donnés annexe C. Nous nous concentrerons ici sur les comparaisons des distributions des temps de détection de dérives entre les expériences menées en échantillonnant le flux d'erreur en utilisant la mesure *ambiguïté* et la mesure aléatoire respectivement. Ainsi, pour chaque jeu de données et chaque méthode de détection de dérive, un test de comparaison Brunner-Munzel est réalisé entre les temps obtenus pour chaque mesure. Il s'agit d'un test non paramétrique généralisant le test de Wilcoxon-Mann-Whitney. Considérant deux variables aléatoires X et Y, l'hypothèse nulle de ce test considérée ici est que $\mathbf{P}(X > Y) \leq 0.5$. Ainsi, l'hypothèse alternative évaluée ici est que, prenant deux temps de détection de dérives obtenus indépendamment, le premier avec la mesure *ambiguïté* et la seconde avec la mesure aléatoire, la

probabilité que le premier temps soit supérieur au second est supérieur a 0.5. L'utilisation de ce test a été choisie à cause de la propension des temps de détection pour certaines méthodes et jeux de données à suivre des distributions très éloignées de lois normales, en particulier en prenant des valeurs grandes avec une grande probabilité. Cela rend notamment les estimations de certaines moyennes et variances instables. C'est également la raison pour laquelle 100 répétitions ont été réalisées pour chaque expérience au lieu de 30 comme précédemment. À titre d'exemple, la figure 3.8 présente les diagrammes quantiles-quantiles des temps de détection de dérives obtenues pour le jeu de données Wine quality et la méthode de détection de dérives Kolmogorov-Smirnov avec $\rho = 0.005$. Le diagramme de gauche correspond au cas ou le flux d'arrivée est échantillonné avec la mesure d'incertitude *Ambiguïté*. Le diagramme de droite correspond au cas ou le flux d'arrivée est échantillonné avec une "mesure d'incertitude" aléatoire. La loi de référence est la loi normale. Ainsi, les distributions des temps de détection ont des profils très différents de lois normales. Les queues à droite des distributions des temps de détection pour les deux stratégies d'échantillonnage sont, en particulier, sensiblement plus épaisses que pour une loi normale.

Les p-values des différents tests de Brunner-Munzel sont présentés table 3.4. Les résultats varient suivant le jeu de données et la méthode d'échantillonnage. Cependant, pour les jeux de données WQ, GT et EG, l'hypothèse nulle peut être rejetée pour 5, 4 et 6 des sept expériences respectivement. Pour ces jeux de données, le biais d'échantillonnage est donc dommageable à la détection de dérives. De plus, pour le méthode de Page-Hinckley avec un seuil haut à $h = 10\sigma$, le test est significatif à 5% sur 4 des six jeux de données (et présente un p-value à 5.4% sur le 6^e. Ainsi, il apparaît que le biais d'échantillonnage présente effectivement un risque de perturber et ralentir la détection de dérives conceptuelles, bien que cette perturbation n'apparaisse pas systématiquement sur tous les jeux de données.

		WQ	AE	CCPP	GT	EG	SD
Kolmogorov-Smirnov	$\rho = 0.005$	$\mathbf{1.9 \times 10^{-6}}$	7.0×10^{-1}	5.5×10^{-1}	$\mathbf{1.4 \times 10^{-3}}$	$\mathbf{1.5 \times 10^{-3}}$	8.0×10^{-1}
	$\rho = 0.001$	1.3×10^{-1}	1.3×10^{-1}	2.0×10^{-1}	$\mathbf{3.2 \times 10^{-4}}$	$\mathbf{8.6 \times 10^{-6}}$	3.1×10^{-1}
Anderson-Darling	$\rho = 0.005$	$\mathbf{5.7 \times 10^{-9}}$	$\mathbf{1.7 \times 10^{-2}}$	4.9×10^{-1}	$\mathbf{8.1 \times 10^{-5}}$	$\mathbf{1.2 \times 10^{-2}}$	6.9×10^{-1}
	$\rho = 0.001$	$\mathbf{1.4 \times 10^{-2}}$	9.3×10^{-1}	2.8×10^{-1}	$\mathbf{4.3 \times 10^{-6}}$	$\mathbf{1.5 \times 10^{-4}}$	9.6×10^{-1}
Page-Hinkley	$h = 10\sigma$	$\mathbf{2.9 \times 10^{-5}}$	3.3×10^{-1}	$\mathbf{5.1 \times 10^{-3}}$	2.2×10^{-1}	$\mathbf{3.1 \times 10^{-5}}$	$\mathbf{2.6 \times 10^{-3}}$
	$h = 15\sigma$	$\mathbf{3.6 \times 10^{-4}}$	6.2×10^{-1}	9.0×10^{-1}	6.1×10^{-1}	$\mathbf{1.6 \times 10^{-7}}$	$\mathbf{1.1 \times 10^{-1}}$
Carte de contrôle		2.9×10^{-1}	$\mathbf{1.1 \times 10^{-2}}$	1.0×10^{-1}	3.9×10^{-1}	1.8×10^{-1}	8.3×10^{-1}

TABLE 3.4 : p-values du test de comparaison de Brunner-Munzel entre les temps de détection de dérives obtenus en échantillonnant le flux en utilisant la mesure d'incertitude *ambiguïté* et les temps obtenus en utilisant la "mesure" aléatoire. Les p-values inférieures à 0.05 sont indiquées en gras

3.6 Conclusion

Ce chapitre introduit la contribution principale de cette thèse, c'est-à-dire une stratégie de couplage entre un modèle de simulation et son métamodèle d'apprentissage automatique dans le contexte des ombres numériques. Le modèle de simulation est, ainsi, supposé avoir un haut niveau de fidélité mais être trop lent pour l'aide à la décision opérationnelle et le traitement de l'intégralité des données collectées continuellement depuis l'homologue physique. Le métamodèle du simulateur est donc introduit pour répondre à cette problématique mais n'est qu'une approximation du modèle de simulation dont l'entraînement reste coûteux en ressources de labélisation. Une stratégie de couplage est donc proposée ici pour bénéficier des avantages respectifs du modèle de simulation et du métamodèle d'apprentissage automatique, plutôt que de remplacer l'un par l'autre.

La stratégie proposée est basée sur le concept d'apprentissage actif dont l'objectif premier est une utilisation efficiente de ressources limitées de labélisation pour l'entraînement d'un modèle d'apprentissage automatique aussi précis que possible. L'objectif premier de la stratégie de couplage introduite ici est, cependant, différent. Il s'agit d'utiliser de manière efficiente les ressources limitées en simulation pour réduire l'erreur préquentielle de toutes les prédictions réalisées par le couple de modèles. La stratégie introduite est également utilisée pour la détection de dérives conceptuelles dans le flux de données collectées auprès de l'homologue physique.

Une étude du comportement du système en régime stationnaire est proposée, et des expériences numériques ont été menées sur huit jeux de données de la littérature scientifique. Un premier ensemble d'expériences compare plusieurs variantes de la stratégie de couplage proposée et les compare à deux stratégies de référence. Il est montré que la stratégie proposée permet effectivement de réduire l'erreur préquentielle du couple tout en requérant moins de simulations qu'une stratégie de maximisation naïve de l'usage du simulateur. Parmi les variantes de la stratégie de couplage étudiée, celle utilisant la mesure *ambiguïté* montre globalement la plus grande réduction de l'EPC.

Le deuxième ensemble d'expériences étudie l'utilisation de la stratégie proposée pour la détection de dérives conceptuelles, et plus particulièrement de dérive dite réelle à partir de flux d'erreurs de prédiction obtenus pour les données dont le vrai label a été obtenu par simulation. Ces expériences sont utilisées pour mettre en lumière une limite de la stratégie proposée. Le biais d'échantillonnage induit par l'échantillonnage actif du flux d'arrivée limite les capacités de détection de dérives des méthodes comparées.

Chapitre 4

Cas des scieries

4.1 Introduction

Les scieries sont des acteurs de la première transformation dans la filière forêt-bois. Elles transforment des grumes et billes de bois (c'est-à-dire des sections de troncs d'arbres) en produits de sciage (poutres, planches...) et autres sous-produits (sciure, copeaux, écorce). Comme illustré figure 4.1, ces sous-produits sont valorisés dans de nombreuses industries telles que l'énergie ou la chimie verte.

L'organisation interne d'une scierie varie d'une entreprise à l'autre. Elles peuvent cependant être généralement décomposées en trois unités de production. La première est l'unité de sciage, qui transforme les grumes et billes de bois initialement stockées dans le parc à grumes en produits semi-finis, ni séchés ni rabotés. Le processus de sciage lui-même est divergent et en coproduction. Ainsi, d'une seule et même grume de bois, une scierie obtient simultanément plusieurs produits de sciage. Par exemple, dans le cas de la figure 4.2, une scierie obtient simultanément deux produits de dimension $5cm \times 8cm$, deux produits de $5cm \times 10cm$ et un produit de dimension $5cm \times 12cm$. Ainsi, le processus de sciage peut être considéré comme un processus de démantèlement (Fletcher et al., 2001). Dans la suite de ce mémoire, nous parlerons d'un panier de produits pour faire référence à la liste des produits de sciage obtenus d'une unique grume ou bille de bois.

Ces produits peuvent ensuite être séchés à l'unité de séchage. Contrairement au processus de sciage qui est une transformation rapide des grumes individuelles en produits et sous-produits, le séchage de ces produits prend plusieurs jours. Il est réalisé dans des séchoirs par lots de produits identiques, ou du moins similaires. La troisième unité de production regroupe les différentes opérations de finition que peuvent subir les produits de sciage. On y retrouve, notamment, le rabotage pour aplanir la surface des produits, et l'application de traitements de surface.

Plusieurs facteurs compliquent significativement la planification des opérations de sciage à court terme. Le fait que le processus de sciage soit en coproduction, notamment, signifie que la réalisation d'une commande spécifique entraîne la production d'autres coproduits de plus faible valeur. De même, la matière première transformée est hétérogène. En effet, toutes les grumes de bois ont des formes différentes, ce qui influence le panier de produits qui en sera obtenu. De plus, chaque grume

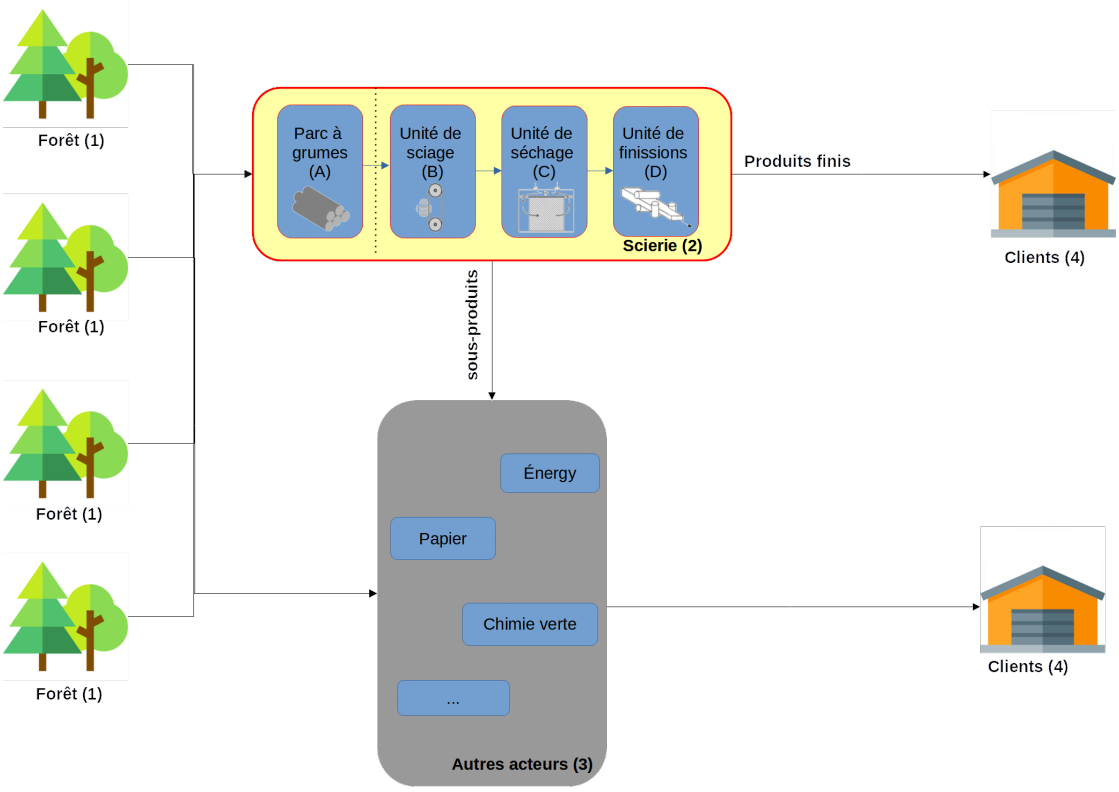

FIGURE 4.1 : Aperçu de la place d'une scierie dans la chaîne logistique de la filière forêt-bois.

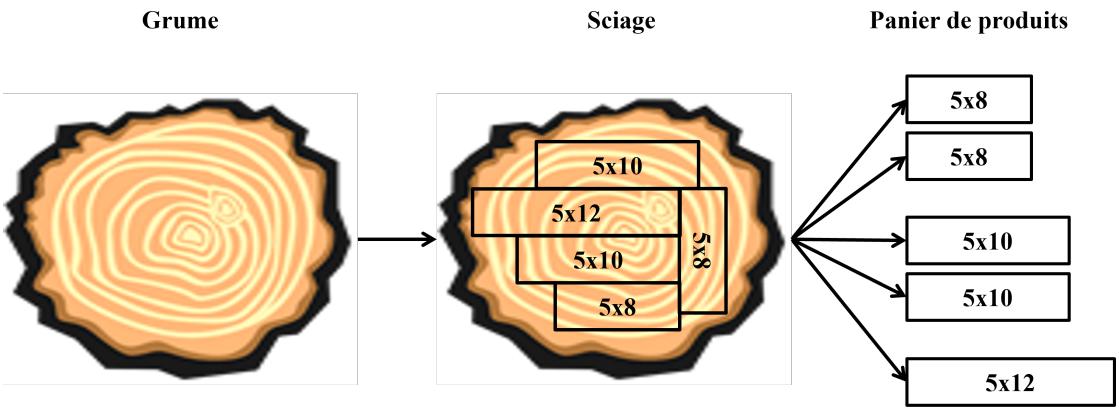

FIGURE 4.2 : Exemple de paniers de produit obtenus après sciage d'une grume, d'après MORIN et al., 2015

de bois possède un ensemble unique de singularités (nœuds, aubiers, pourriture...) qui peuvent influencer la qualité des produits finis. Ces singularités sont souvent impossibles à observer avant ouverture des grumes, sauf à utiliser des moyens de contrôle non destructif coûteux comme des scanners à rayon X.

Comme toute industrie sujette à une compétition internationale intense, les scieries se doivent d'optimiser leurs processus de production. Shahi, 2016, par exemple, évoque le cas de l'industrie canadienne qui a souffert, ces dernières années, des effets combinés de la mondialisation, de changements dans le marché de la construction, et de fluctuations des taux de changes. En France, Chalayer, 2017 estime qu'une scierie ferme en moyenne tous les trois jours.

Ce chapitre a pour objectif l'identification des bénéfices que peuvent apporter les Ombres et Jumeaux numériques (O&JN) à l'industrie du sciage. Les défis à relever et barrières à leurs mises en place seront également considérés. Nous nous concentrerons, en particulier, sur les O&JN pour l'aide à la planification et au contrôle de la production. Nous considérerons donc, par défaut, des *instances d'O&JN* plutôt que des *O&JN prototype*, c'est-à-dire que nous considérerons toujours l'existence d'un homologue physique. De plus, la phase du cycle de vie considérée, que ce soit pour les produits ou processus, est celle de la production indiquée dans la figure 2.4.

L'usage et les limites des O&JN dans le cadre d'opérations de production industrielles ont été étudiés dans de nombreux articles récents de la littérature scientifique (Melesse, Pasquale et Riemma, 2020 ; Santos et al., 2021 ; Zheng, Lu et Kiritsis, 2021). Cependant, comme évoqué dans le chapitre 2, la définition et les services rendus par les O&JN sont souvent spécifiques à leur domaine d'application. Il est donc nécessaire d'analyser le potentiel O&JN dans le cadre des scieries. De même, bien qu'il existe plusieurs études dans la littérature scientifique sur les effets des technologies de l'industrie 4.0 sur la chaîne d'approvisionnement des produits forestiers (Müller, Jaeger et Hanewinkel, 2019), ainsi que sur l'utilisation de l'optimisation et de la recherche opérationnelle dans l'industrie des produits forestiers (D'Amours, Rönnqvist et Weintraub, 2008), aucune ne se concentre sur l'utilisation et les avantages potentiels des O&JN pour l'industrie du sciage en particulier. En effet, elles se concentrent soit sur les opérations forestières, soit sur l'utilisation d'autres technologies. Ce chapitre présente les résultats d'une revue de littérature sur les problématiques de planification et de contrôle de la production en scierie, et discute ces résultats sous l'angle des O&JN. Cette revue ne se concentre pas directement sur les articles traitant de l'utilisation d'O&JN en scierie car très peu d'articles traitent du sujet.

La suite de ce chapitre est organisée comme suit. La section 4.2 présente la méthodologie suivie pour le travail de littérature présenté dans ce chapitre. La section 4.3 présente les problématiques de planification communément rencontrée dans la littérature et les solutions apportées. La section 4.4 discute les applications envisagées pour les $O\&JN$ en scierie et fait le lien avec la contribution de ce mémoire. Enfin, la section 4.6 conclut ce chapitre.

4.2 Méthodologie

Pour cette revue de littérature, nous avons utilisé la méthode de sélection systématique par effet boule de neige. Le principe de cette méthode est comme suit. D'abord, un premier ensemble d'articles est sélectionné dans les bases d'articles scientifiques usuelles. Ici, nous utilisons IEEEXplore, Web of science, ScienceDirect et Scopus. Cet ensemble est ensuite complété par étapes

successives de sélection avant et arrière jusqu'à ce qu'un nombre suffisant de documents soient collectés. L'objectif n'est pas ici d'effectuer une revue exhaustive de tous les documents traitant d'un sujet, mais d'en obtenir un échantillon représentatif. L'échantillonnage avant consiste en la sélection de documents référencés dans Google Scholar comme citant des documents précédemment sélectionnés. De même, l'échantillonnage arrière consiste en la sélection d'articles référencés dans les bibliographies de documents précédemment sélectionnés. Les documents obtenus ainsi sont d'abord filtrés par titre, puis par résumé et mots-clés comme recommandé par Wohlin, 2014. Enfin, une dernière sélection est réalisée à la lecture des articles complets. Ici, seules une étape de sélection avant et une étape de sélection arrière ont été réalisées.

L'ensemble initial de documents constitué pour cette revue contient 50 articles provenant des bases d'articles scientifiques IEEEXplore, Web of science, ScienceDirect et Scopus. La chaîne de recherche utilisée est *'sawmill' and ('planning' or 'production control' or 'scheduling' or 'sequencing')*. Dans ScienceDirect, ces termes ont été cherchés uniquement dans les résumés et mots-clés donnés par les auteurs. Dans Web of Science, la recherche considère également le champ Keyword Plus qui contient des mots-clés supplémentaires attribués par la base elle-même. De même, la recherche dans IEEEXplore, considère également les termes d'indexation. Le reste de la sélection par effet boule de neige a été réalisé avec Google Scholar. Au total, 133 documents étudiant des problèmes de planification de la production en scierie ont été collectés ainsi. Les plus représentatifs sont détaillés dans la suite de ce chapitre. Seuls des documents en langue anglaise ont été considérés. Cependant, aucune limitation n'a été placée sur la date de publication. De plus, des documents de différents types sont considérés : communications en congrès, articles de revues scientifiques, mémoires de thèses et mémoires de master. Ces documents ont uniquement été considérés, cependant, si le texte intégral était accessible en ligne.

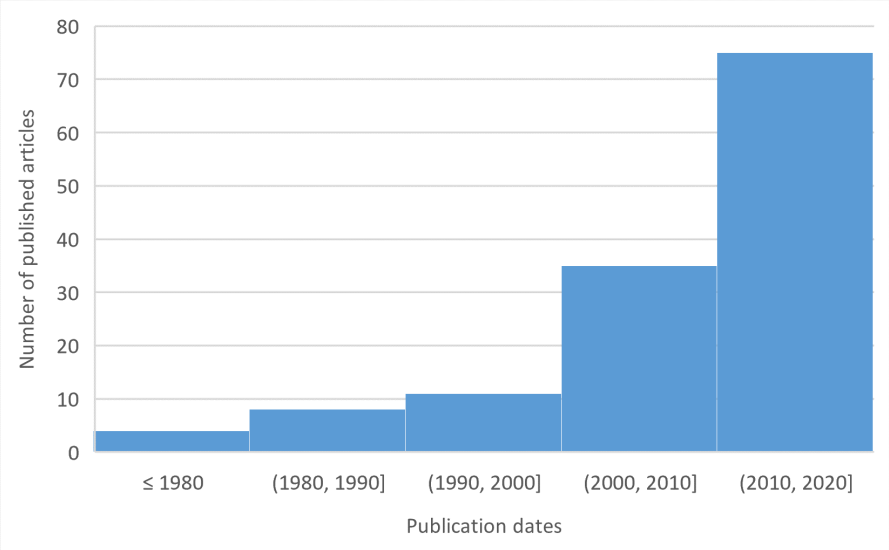

Figure 4.3 : Histogramme des dates de publication des documents collectés par effet boule de neige

La figure 4.3 présente l'histogramme des dates de publication des documents collectés. Malgré l'absence de filtrage par date des documents collectés, la plupart sont relativement récents. En particulier, 75 des documents collectés ont été publiés après 2011.

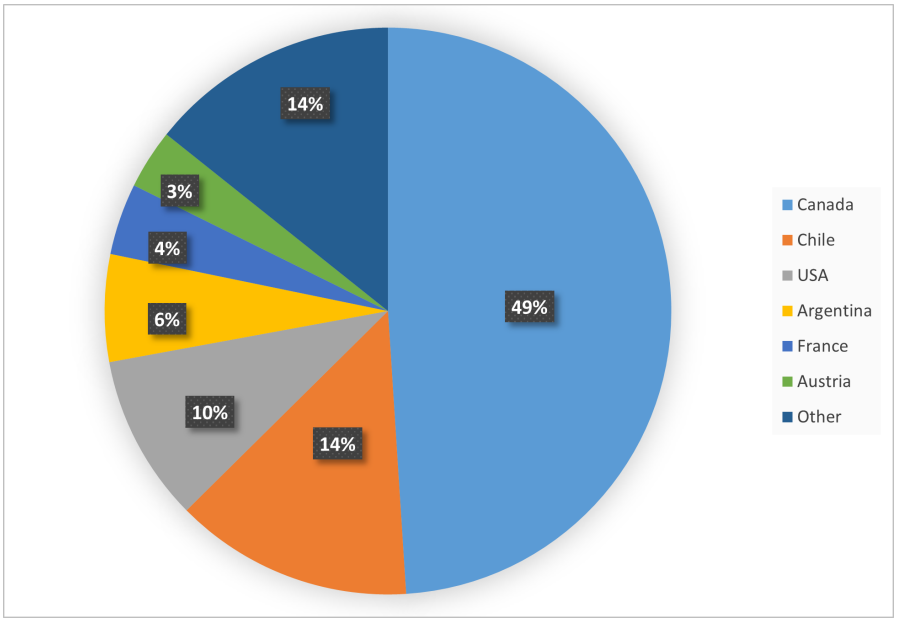

FIGURE 4.4 : Diagramme circulaire des pays d'appartenance des institutions des auteurs des documents collectés durant ce travail de revue.

la figure 4.4 détaille les pays d'appartenance des institutions des auteurs des documents collectés. Ce champs de recherche est largement dominé par les institutions canadiennes qui ont contribué à presque la moitié des documents collectés. Le fait que leur gouvernement fédéral, provinces et territoires possèdent 90% des forêts canadiennes et gèrent la distribution des ressources aux acteurs industriels est sans doute un des facteurs expliquant l'importance de ce secteur de recherche dans leur pays.

Les mots-clés des articles sélectionnés ont été rassemblés et résumés sous la forme d'un graphe de cooccurrence à l'aide du logiciel VOSviewer, puis remodelés à l'aide de Gephi pour plus de clarté. Les résultats sont présentés dans la figure 4.5. Seuls les mots-clés présents dans plus de trois articles sont présentés. En outre, l'orthographe des mots-clés a été homogénéisée. En particulier, l'orthographe américaine a été utilisée, et les caractères spéciaux tels que '-' ont été supprimés. Un lien entre deux nœuds indique la cooccurrence des mots-clés associés dans un article. L'épaisseur d'un lien est proportionnelle au nombre d'articles dans lesquels les mots-clés liés sont cooccurrents. La taille d'un nœud correspond au nombre d'articles dans lesquels le mot-clé associé apparaît. Comme on peut le constater, de nombreux mots-clés correspondent soit à des termes liés au terrain, soit à des méthodes de résolution de problèmes de planification. Par exemple, "sawmill", "cutting pattern" et "forestry" sont tous des mots-clés spécifiques à cette industrie. De même, même s'ils ne sont pas spécifiques à l'industrie des produits forestiers, "production planning", "supply chain management" et "tactical planning" sont des termes attendus du type de problème considéré. En ce qui concerne les méthodes de résolution de problèmes, les mots clés "optimization" et "simulation" sont fréquents. Ils apparaissent dans 17 et 14 articles respectivement. À titre de comparaison, "sawmill" et "production planning", qui sont attendus au vu de la chaîne de recherche initiale, sont des mots-clés dans 22 et 16 articles respectivement. Les méthodes d'optimisation spécifiques sont également fréquentes. Le mot-clé "linear programming" est inclus

dans 11 articles, et "mixed integer programming" dans 7.

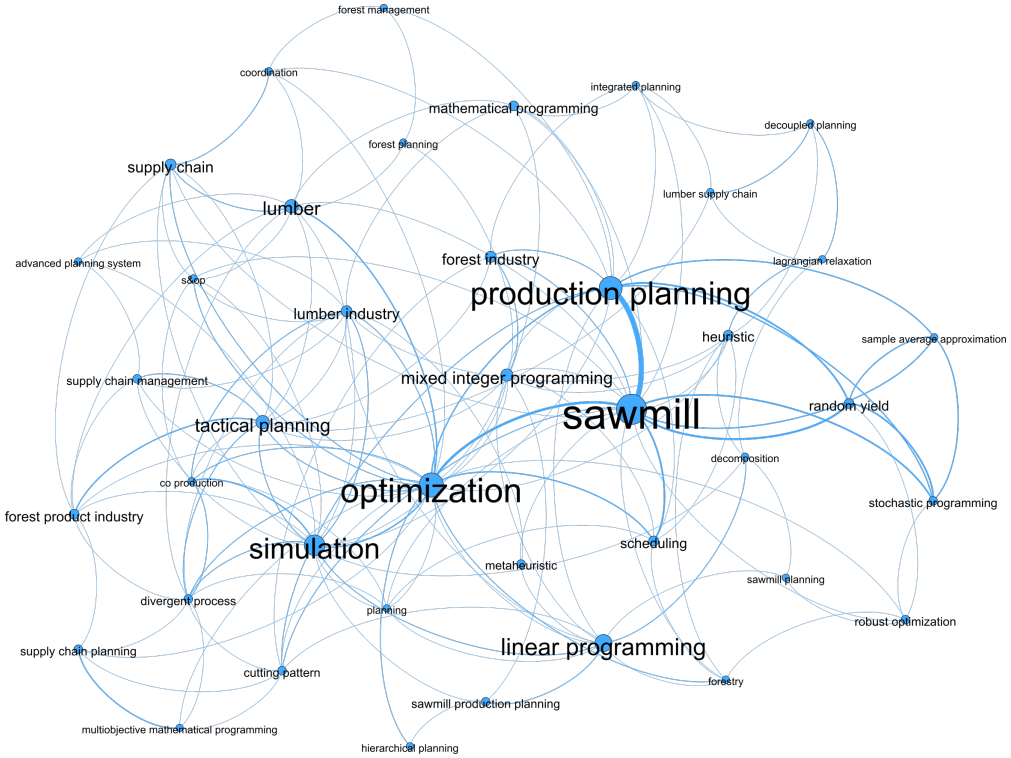

FIGURE 4.5 : Graphe des mots-clé des articles sélectionnés

4.3 Planification de la production en scierie

4.3.1 Niveaux de planification

La planification dans l'industrie forêt-bois peut être divisée en quatre niveaux hiérarchiques : stratégique, tactique, opérationnel et en ligne (RÖNNQVIST, 2003 ; D'AMOURS, RÖNNQVIST et WEINTRAUB, 2008).

Le niveau stratégique fait référence à des problèmes dont l'horizon de planification est long, généralement supérieur à cinq ans. Des exemples de problèmes stratégiques dans l'industrie du sciage concernent la création de nouvelles usines, le choix de leur emplacement, ou l'expansion d'une usine existante. TRONCOSO et GARRIDO, 2005 ont proposé l'utilisation d'un modèle de programmation mixte en nombres entiers pour résoudre trois problèmes stratégiques de ce type dans l'industrie forestière. Les auteurs étudient notamment l'optimisation de l'emplacement d'unités de productions.

La planification tactique concerne les opérations à moyen terme. RÖNNQVIST, 2003 les classent comme ayant un horizon temporel de six mois à cinq ans. Cependant, certains auteurs présentent

des problèmes tactiques avec des horizons temporels plus courts. Par exemple, Lobos et Vera, 2016 considèrent un problème de niveau tactique avec un horizon roulant de quatre mois (l'optimisation de la quantité mensuelle de grumes à acheter et de main-d'œuvre à employer). Un autre exemple de problème de planification tactique est l'optimisation du plan industriel et commercial Marier et al., 2014. Comme souligné dans D'Amours, Rönnqvist et Weintraub, 2008, une caractéristique importante des plans tactiques est qu'ils servent de pont entre les décisions stratégiques et les plans opérationnels.

Le niveau opérationnel implique des problèmes de décision à court terme. Rönnqvist, 2003 classe ces problèmes comme ayant un horizon temporel inférieur à six mois. D'Amours, Rönnqvist et Weintraub, 2008 ont également souligné la nécessité pour ces problèmes de considérer des périodes de planification très courtes, de quelques heures à un jour au maximum, afin d'être aussi proche que possible des opérations sur le terrain. La planification de la production quotidienne (Zanjani, Ait-Kadi et Nourelfath, 2010) et l'ordonnancement des opérations de séchage (Huka, Rindler et Gronalt, 2020) sont des exemples de problèmes opérationnels.

Enfin, le niveau en ligne concerne le processus de production immédiat et la réalité opérationnelle. Rönnqvist, 2003 a classé ces problèmes de planification comme ayant un horizon de moins d'un jour, par exemple la gestion des arrivées de camions grumier en scierie ou le contrôle de processus. Todoraki et Ronnqvist, 2001 présentent un exemple d'un tel problème. Cet article propose une solution pour ajuster en ligne la liste des prix des produits en fonction de la production réelle. Cette liste de prix est utilisée par un optimiseur pour décider quel plan de coupe utiliser pour quelle grume. L'objectif est ici de remplir une liste de commandes en minimisant la production de produits non désirés.

4.3.2 Défis rencontrés

Plusieurs facteurs compliquent la planification opérationnelle dans l'industrie forêt-bois en général et dans l'industrie du sciage en particulier. Le premier défi découle de la nature du processus de sciage mentionnée dans l'introduction du chapitre. Le processus de sciage est divergent et en coproduction, ce qui implique que produire un ensemble de produits pour répondre à une commande entraîne également la production d'autres produits indésirables de faible valeur.

Une deuxième difficulté réside dans les incertitudes inhérentes au processus de production de la scierie. En plus de l'incertitude sur la demande, les scieries doivent tenir compte des incertitudes sur l'offre en matière première et le mix produit. Plusieurs facteurs rendent plus difficile la prévision de la production des scieries, que ce soit pour une seule ou pour plusieurs grumes, comme l'hétérogénéité de la matière première, et la présence d'optimiseurs en temps réel intégrés dans certaines machines de scierie. Ces optimiseurs sont des programmes encapsulés dans des machines qui choisissent le plan de coupe de chaque grume en temps réel afin de maximiser la valeur ou le volume des produits obtenus. Ils effectuent une optimisation locale ou peuvent considérer une projection simplifiée des étapes de traitement futures. Ces optimiseur locaux compliquent encore la prévision en permettant une configuration supplémentaire de l'équipement et en ajoutant d'autres variables dans le processus, comme la liste des valeurs des produits utilisée durant l'optimisation. S'ils permettent effectivement de maximiser la valeur obtenue à partir de la matière première disponible, ils compliquent donc la planification. L'effet exact du changement de configuration des machines sur l'ensemble de la production est en effet difficile à évaluer, car cela modifie le panier de produits sciés dans chaque grume. L'impact négatif est particulièrement

important en l'absence de données historiques, par exemple lorsque de nouveaux produits sont introduits (Wery et al., 2014 ; Wery et al., 2018).

Une autre difficulté est l'intégration des opérations de sciage dans la chaîne d'approvisionnement de la filière forêt-bois dans son ensemble. En particulier, les forêts, qui fournissent les grumes aux scieries, doivent, en général, être gérées sur des horizons extrêmement longs pour prendre en compte au moins une rotation forestière complète. Une telle rotation forestière peut prendre plus de 80 ans dans les pays froids comme le Canada (D'Amours, Rönnqvist et Weintraub, 2008). De même, la gestion forestière tactique peut, par exemple, considérer un horizon de 5 à 10 ans. Comme souligné par Marinescu et Maness, 2010, les gestionnaires de scieries peuvent avoir des difficultés à intégrer ces données dans leurs modèles de planification car ils considèrent des horizons temporels beaucoup plus courts. De plus, l'accumulation de données sur une période de 80 ans pose ses propres défis.

En plus des objectifs économiques, il est intéressant pour l'industrie forêt-bois d'intégrer des aspects sociaux et environnementaux dans le processus de planification des opérations. Cela inclut bien sûr les activités des scieries. En effet, elles sont un acteur important de la gestion forestière, qui fournit des services sociaux et écologiques à l'ensemble de la société, en termes de préservation de la faune et de la flore, de visites touristiques et de séquestration du carbone. En outre, les scieries sont des employeurs importants dans certaines zones rurales. Parmi tous les articles recueillis, seuls cinq ont présenté de tels objectifs. Par exemple, Boukherroub et al., 2013b ; Boukherroub et al., 2013a ; Boukherroub et al., 2015 proposent des modèles basés sur des techniques d'optimisation multi-objectifs pour optimiser simultanément des indicateurs économiques, environnementaux et sociaux, tels que les coûts d'exploitation, les émissions de gaz à effet de serre et l'emploi local. Dumont et al., 2019 ont, de même, proposé une méthode pour intégrer les coûts de consommation d'énergie dans la planification tactique de la production en scieries en adaptant un modèle existant. Plus précisément, ils ont adapté le modèle de programmation linéaire introduit par Marier, Gaudreault et Robichaud, 2014 pour inclure le coût énergétique des processus de production et le chauffage ou le refroidissement des lieux de travail dans la fonction objectif minimisée.

Malgré les difficultés engendrées, l'intérêt de coordonner plusieurs éléments de la chaîne d'approvisionnement externe et interne d'une scierie est souvent souligné dans la littérature. En particulier, Rönnqvist et al., 2015 considèrent la gestion intégrée de la chaîne logistique forêt-bois comme un problème ouvert dans l'industrie. Plusieurs études ont démontré l'avantage d'une telle coordination. Par exemple, Bajgiran, Zanjani et Nourelfath, 2016 proposent un modèle de programmation mixte en nombres entiers pour intégrer simultanément les activités de récolte, d'approvisionnement, de production et de vente dans la chaîne d'approvisionnement en bois d'œuvre. Les résultats de la méthode proposée sont comparés à ceux obtenus à partir de modèles découplés, et une diminution de 11% du bénéfice a été observée lorsque la planification est découplée en deux modèles. Une diminution de 84% est observée en la découplant en trois modèles. De même, Troncoso et al., 2015 comparent un modèle découplé dans lequel les opérations de la forêt et de l'usine sont planifiées de manière séquentielle, et ou la planification des acteurs forestiers vise uniquement à maximiser la valeur actuelle nette du bois, avec un modèle intégré dans lequel les deux solutions visent à maximiser le profit à long terme de l'entreprise. Par rapport au modèle découplé, qui est présenté comme la stratégie de planification traditionnelle dans la chaîne de valeur forestière, le modèle intégré permet à l'entreprise de réaliser un bénéfice plus élevé malgré la récolte de moins d'arbres. Si les deux articles démontrent l'intérêt de l'intégration forêt-scierie pour la planification dans la chaîne de valeur, ces deux modèles diffèrent grandement

en termes d'horizon de planification. Bajgiran, Zanjani et Nourelfath, 2016 ne planifient les opérations que sur un horizon de planification annuel, ce qui est trop court pour prendre en compte les opérations de sylviculture, et l'étude se concentre sur les opérations de récolte, tandis que Troncoso et al., 2015 planifient les opérations forestières sur au moins une rotation complète. Cependant, les opérations en aval ne sont planifiées que sur cinq ans car la demande devient trop incertaine après cet horizon. Enfin, plusieurs articles examinent le problème du tronçonnage des tiges d'arbres en grumes. Si ce problème peut survenir aussi bien dans les forêts que dans les scieries, il s'agit d'une étape divergente qui affecte fortement la production ultérieure et les bénéfices des scieries (Dems, Rousseau et Frayret, 2015 ; Vanzetti et al., 2019b).

Un défi important, mais rarement mentionné dans les articles sélectionnés, est la traçabilité des grumes et des produits de la forêt aux scieries et tout au long du processus de sciage. La nature divergente du processus de sciage rend les marquages discrets communément utilisés dans l'industrie tels que les étiquettes RFID inadaptées pour suivre l'origine de chaque bois (Jover, Thomas et Bombardier, 2011). Néanmoins, Boukherroub et al., 2015 soulignent l'existence d'une forte demande pour une telle méthode de traçabilité des produits afin de mieux suivre l'origine du bois et de calculer l'empreinte carbone. La traçabilité est également motivée par des réglementations gouvernementales. Par exemple, le règlement européen sur le bois, qui vise à lutter contre l'exploitation illégale des forêts. La traçabilité des grumes et des produits du bois est toutefois rendue encore plus complexe par la nature divergente du processus de sciage.

4.3.3 Problèmes de planification couramment rencontrés dans la littérature

La littérature présente un large éventail de problèmes de planification présents dans les scieries, allant de la création de plans de production spécifiques aux différentes unités de production individuelles jusqu'à l'intégration verticale ou horizontale des plans à différents niveaux. Un problème courant est la planification des opérations primaires de sciage. Certains auteurs parlent de *cutting pattern problem* (Vergara, Palma et Sepúlveda, 2015 ; Palma et Vergara, 2016) ou de *sawing stock problem* (Pradenas et al., 2013). Une formulation générale de ce problème est "quelle quantité de quelle classe de grumes scier avec quel plan de coupe pour optimiser un objectif donné sous contraintes". Cependant, les objectifs et les contraintes considérés varient d'un cas à l'autre. Des objectifs courants sont la maximisation des profits ou la minimisation des coûts. Par exemple, Kazemi Zanjani, Nourelfath et Aït-Kadi, 2011 proposent un modèle de programmation stochastique pour résoudre ce problème sous un horizon de planification de 30 jours avec un objectif de minimisation des coûts. Huka et Gronalt, 2017 proposent un algorithme de programmation mixte en nombres entiers pour maximiser le revenu net de l'opération sur un horizon de planification de 10 semaines. Vanzetti et al., 2018 proposent, de même, un modèle de programmation linéaire mixte en nombres entiers pour maximiser le bénéfice net sur un horizon de planification de 5 jours. De plus ils proposent une étape préliminaire qui génère les plans de coupe utilisés dans le problème d'optimisation.

D'autres auteurs proposent des variations de ce problème. Par exemple, dans Haberl et al., 1991 ; Maturana, Pizani et Vera, 2010 ; Varas et al., 2014, les grumes d'une certaine classe sont systématiquement coupées suivant le plan de coupe qui maximise le rendement attendu sur ladite classe. En particulier, ils n'optimisent que le mélange de matières premières à traiter pour répondre à une commande. Ils justifient cette approche comme un moyen pour les scieries de valoriser au mieux la matière première. À l'inverse, des auteurs tels que Wery et al., 2018 ;

Maturana, Pizani et Vera, 2010 rapportent que certaines scieries ont automatisé le sciage des grumes au point que le plan de coupe utilisé sur chaque grume spécifique est décidé en temps réel par le matériel pour maximiser le volume ou la valeur en sortie. Cela limite les actions du gestionnaire au choix du mélange de grumes à traiter, à la configuration de la scierie et à la liste des prix des produits utilisés par le matériel lorsqu'une optimisation par valeur est effectuée.

Les étapes suivantes du processus de production, la gestion des approvisionnements, des stocks et des ventes peuvent également être incluses dans ces modèles de planification avec différents niveaux de précision pour planifier l'ensemble de la production d'une ou plusieurs scieries.

Par rapport au sciage, moins d'exemples portent sur la planification et l'ordonnancement des opérations de séchage et de finition. Par exemple, Huka, Rindler et Gronalt, 2020 proposent des méthodes d'ordonnancement permettant de minimiser les retards des opérations de séchage. Vanzetti, Corsano et Montagna, 2020 proposent un modèle de programmation en logique disjonctive pour planifier ces opérations sur 5 jours afin de maximiser le nombre de lots traités ou de minimiser l'espace inutilisé. Marier, Gaudreault et Noguer, 2021 proposent de combiner la programmation mixte en nombres entiers et la programmation par contraintes pour minimiser les retards sur un horizon de planification de trois à quatre semaines. Theresia, Widyadana et Wahjudi, 2019 proposent deux modèles de programmation linéaire pour affecter successivement les bois de construction à différents séchoirs puis à différents compartiments des séchoirs afin de minimiser les coûts de production et l'utilisation des compartiments.

Comme pour les opérations de séchage et de finition, le stockage des grumes et billes de bois dans le parc à grumes de la scierie présente son propre ensemble de problématiques. Par exemple, Beaudoin, LeBel et Soussi, 2012 proposent l'utilisation d'un modèle de simulation à événements discrets pour comparer les stratégies d'affectation des chargeurs aux camions et optimiser les opérations de déchargement. Rathke, Huka, Gronalt et al., 2013, quant à eux, étudient le problème de l'optimisation des opérations de tri et de stockage des grumes afin de minimiser le temps de transport.

En général, la coordination des différentes étapes de production peut être divisée en deux catégories : centralisée ou distribuée. Lorsque la planification est effectuée de manière centralisée, un seul schéma d'optimisation, généralement un modèle unique, est proposé pour optimiser toutes les opérations simultanément. Néanmoins, Gaudreault et al., 2010 défendent que les modèles de planification centralisés sont souvent mal adaptés à la production de plans opérationnels détaillés en raison de la complexité des opérations sous-jacentes. L'alternative, selon les auteurs, est de favoriser une approche distribuée ainsi que des mécanismes de coordination entre les différentes unités de production.

Parmi les exemples de plans centralisés, on peut citer Vanzetti et al., 2019a qui proposent un modèle de programmation linéaire mixte en nombres entiers pour maximiser le profit d'une scierie à partir du sciage primaire des grumes en produits intermédiaires et des opérations de sciage secondaires en produits finaux, qui a lieu après les opérations de séchage. De même, Gaudreault et al., 2011 proposent deux approches, basées soit sur la programmation mixte en nombres entiers, soit sur la programmation par contraintes pour la planification et l'ordonnancement des opérations de séchage et de finition.

Plusieurs études proposent, au contraire, des plans décentralisés. En particulier, le Consortium de recherche FORAC (Université Laval, QC, Canada) a développé une plateforme de simulation basée sur des systèmes multi-agents pour modéliser l'interaction entre les différentes unités de

production, les fournisseurs et les clients. Cette plateforme est décrite dans Frayret et al., 2007. Un large éventail de problèmes a été étudié sur cette plateforme. Par exemple, Cid Yáñez et al., 2009 modélisent des stratégies de coordination avec différents points de découplage entre flux tirés et poussés. Il est intéressant de noter qu'ils concluent que si le déplacement de ce point de découplage en amont de la chaîne d'approvisionnement interne augmente la satisfaction du client, la nature divergente du processus de production nuit à la capacité de la scierie à maximiser le rendement en valeur des grumes. Gaudreault et al., 2010 proposent plusieurs mécanismes de coordination entre agents de planification de la production et les testent sur la plateforme. Ils comparent notamment la planification en amont, la planification en deux phases et la planification par goulet. La planification en amont est un système en tirage pur dans lequel les agents planifient leur production de manière itérative en commençant par l'agent le plus proche du client et en propageant la demande vers l'aval de la chaîne d'approvisionnement. Dans la planification en deux phases, un premier tour de planification est effectué, avec des informations sur la demande qui se propagent vers l'amont, jusqu'au premier agent de production, c'est-à-dire l'agent de sciage. Puis, dans le deuxième tour, ces agents propagent leur plan de production en aval, en les adaptant aux informations sur la demande. La planification par goulet est similaire à la planification en deux phases, mais le premier tour ne part que de l'agent de séchage, qui est considéré comme un goulet dans le processus en raison du temps nécessaire au séchage (Dumetz et al., 2017).

4.3.4 Technologies d'aide à la décision

Deux technologies d'aide à la décision couramment rencontrées sont les modèles d'optimisation et de simulation.

Les méthodes d'optimisation sont largement utilisées dans l'industrie forêt-bois depuis des décennies (D'Amours, Rönnqvist et Weintraub, 2008 ; Rönnqvist et al., 2015). La littérature propose de nombreux cas d'utilisation de ces approches pour résoudre des problèmes de planification en scierie. Par exemple, Donald, Maness et Marinescu, 2001 proposent l'utilisation de modèles de programmation linéaire pour optimiser la planification mensuelle de la production dans une scierie. Gaudreault et al., 2011 comparent un modèle de programmation mixte en nombres entiers avec un modèle de programmation par contraintes pour l'ordonnancement des opérations de séchage et de finition. Considérant la taille importante du problème industriel qu'ils étudient, ils proposent en outre une procédure de recherche pour trouver une solution satisfaisante en un temps limité. Ce problème de temps de calcul limité est souligné de manière similaire par Marier, Gaudreault et Robichaud, 2014, qui relatent l'implémentation d'un modèle d'optimisation mixte en nombres entiers pour un partenaire industriel. Une exigence forte était que le modèle puisse proposer une solution en moins de cinq minutes.

Les méthodes de résolution basées sur des métaheuristiques pour résoudre ces problèmes d'optimisation semblent prometteuses. Les métaheuristiques sont une famille générale de méthodes pouvant être utilisées pour trouver des solutions approchées à des problèmes complexes dans un temps acceptable (Chopard et Tomassini, 2018). En particulier, de telles méthodes sont considérées dans la littérature lorsque le problème n'est pas convexe et/ou difficile à résoudre. Par exemple, Huka, Rindler et Gronalt, 2020 proposent plusieurs heuristiques pour résoudre un problème de programmation non linéaire visant à planifier des opérations de séchage. Ils proposent notamment l'utilisation d'une métaheuristique (une recherche tabou, (Laguna, 2018)) pour améliorer les résultats de l'heuristique. De même, Caballero et al., 2009 proposent l'utilisation d'une métaheuristique pour résoudre un modèle de programmation par objectifs non linéaires

pour planifier la production d'une scierie et répondre à la demande. La métaheuristique utilisée est un algorithme génétique pour l'optimisation multi-objectif nommée *Scatter Search Procedure for Multiobjective Optimization* (Molina et al., 2007). La table 4.1 présente un résumé des métaheuristiques et des caractéristiques des problèmes pour lesquels elles sont considérées dans les articles étudiés.

Référence	**Problème étudié**	**Métaheuristique**		
		Recherche avec Tabou	**Algorithme évolutionnistes**	**Recuit-simulé**
Pradenas, Peñailillo et Ferland, 2004	Optimisation des opérations mensuelles. Le problème est modélisé par un MIP avec des contraintes quadratiques.	X		
Caballero et al., 2009	Planification opérationnelle, modélisée comme un multi-objectif non linéaire à variables entières.		X	
Lindner, Visser et Wessels, 2013	Optimisation simultanée des opérations de sciage primaires et secondaires.		X	
Pradenas et al., 2013	Planification des opérations de sciage sur plusieurs jours. Les métaheuristiques sont utilisées pour générer des plans de coupe.		X	X
Baesler et Palma, 2014	Ordonnancement des opérations de finition sur un groupe de machines travaillant en parallèle.		X	

Lindner, Vlok et Wessels, 2015	Optimisation simultanée des réglages de plusieurs machines.		X	
Shahi et Pulkki, 2015	Optimization d'une chaine logistique dans la filière forêt-bois.	X	X	
Huka, Rindler et Gronalt, 2020	Ordonnacement des opérations de séchage. La métaheuristique est initialisée à partir d'une première heuristique spécifique au problème posé. Le problème est modélisé comme une MIP avec des contraintes non linéaires.	X		

Table 4.1 : Articles collectées durant la revue de littérature proposant l'utilisation de métaheuristiques.

Enfin, la dernière utilisation des modèles d'optimisation est la création de plans tenant compte d'une ou plusieurs sources d'incertitudes. Selon Zanjani, Nourelfath et Ait-Kadi, 2013b, une méthode couramment utilisée dans l'industrie pour faire face à ces incertitudes est l'utilisation de modèles d'optimisation déterministes mais utilisant un horizon roulant. Le modèle d'optimisation est résolu pour un horizon long en utilisant les informations disponibles à ce moment-là. Cependant, le plan n'est pas suivi jusqu'à la fin de l'horizon de planification, mais est recalculé régulièrement lorsque de nouvelles informations deviennent disponibles. Cependant, de nombreuses études proposent d'utiliser des modèles qui prennent explicitement en compte les incertitudes. Les deux méthodes les plus populaires sont la programmation stochastique et l'optimisation robuste. Ces deux méthodes ont leurs propres avantages et inconvénients. La table 4.2 présente un résumé des articles utilisant ces méthodes recueillies lors de la revue de la littérature. En particulier, Varas et al., 2014 soulignent le fait que la programmation stochastique nécessite la connaissance de la distribution de probabilité des événements, qui peut ne pas être facile à rassembler ou à estimer. De plus, ces méthodes peuvent être difficiles à comprendre pour un gestionnaire de scierie et peuvent être intensives en calcul selon le nombre de scénarios considérés. Diverses méthodes de décomposition et heuristiques ont donc été proposées (par exemple, Zanjani, Nourelfath et Ait-Kadi, 2013a ; Zanjani, Ait-Kadi et Nourelfath, 2013). Ces mêmes auteurs ont également utilisé des méthodes de simulation et d'estimation Monte Carlo pour générer des scénarios (Zanjani, Nourelfath et Ait-Kadi, 2007). Alors que l'optimisation stochastique nécessite la modélisation de la distribution de probabilité sur les scénarios, l'optimisation robuste vise à fournir un plan réalisable malgré l'incertitude sur certains des paramètres du modèle et ne nécessite pas une telle modélisation des probabilités. Cependant, Rönnqvist et al., 2015 soulignent que cela peut conduire à des plans conservateurs aux coûts élevés.

Référence	Type d'incertitude				Solution étudiée	
	Matière première	Production	Demande	Autre	Programmation stochastique	Optimisation robuste
Vila, Beauregard et Martel, 2009			X		X	
Zanjani, Ait-Kadi et Nourelfath, 2010		X				X
Kazemi Zanjani, Nourelfath et Ait-Kadi, 2010	X		X		X	
Kazemi Zanjani, Nourelfath et Aït-Kadi, 2011	X				X	
Vahidian, 2012		X	X		X	
Zanjani, Nourelfath et Ait-Kadi, 2013b		X	X		X	X
Zanjani, Ait-Kadi et Nourelfath, 2013		X	X		X	
Kazemi Zanjani, Ait-Kadi et Nourelfath, 2013		X			X	
Zanjani, Nourelfath et Ait-Kadi, 2013a		X	X		X	

Varas et al., 2014	X		X			X
Alvarez et Vera, 2014		X				X
Palma et Vergara, 2016				préférences managériales		X
Lobos et Vera, 2016	X				X	
Alvarez et al., 2020		X				X

Table 4.2 : Articles implémentant des méthodes d'optimisation spécifiques pour la prise en compte des incertitudes.

La simulation devient, quant à elle, un outil de plus en plus accepté et important dans l'industrie et est largement utilisée dans l'industrie forestière, par exemple, pour identifier les goulets d'étranglement ou étudier l'impact de la mise à niveau des systèmes de production (Opacic, Sowlati et Mobini, 2018). Notamment, Masood et Sonntag, 2020 classent la simulation comme une technologie de l'industrie 4.0 présentant en moyenne un bénéfice élevé pour une faible complexité d'application, ce qui la rend particulièrement adaptée aux petites et moyennes entreprises. L'industrie des scieries a à sa disposition plusieurs simulateurs de sciage, qui peuvent servir pour l'aide à la décision aux niveaux tactique et stratégique. Un intérêt particulier de ces simulateurs est qu'ils peuvent remplacer les données historiques de production lorsqu'elles ne sont pas disponibles, par exemple, lors de la conception d'une nouvelle scierie ou du traitement de produits inhabituels. Ils peuvent également produire des données pour des grumes individuelles, atténuant ainsi les incertitudes liées à leur transformation.

Ces simulateurs de sciage ont été utilisés dans de nombreuses études pour générer des données qui sont utilisées pour résoudre des problèmes d'optimisation. Par exemple, Maness et Norton, 2002 utilisent un simulateur pour simuler le sciage de grumes en se basant sur des scans aux rayons X. Certains simulateur intègrent, également, des modules d'optimisation de plans de coupe Sohrabi, 2013. Ces données sont ensuite intégrées dans d'autres modèles pour l'optimisation de plans de production. Toutefois, cette approche a été critiquée par Zanjani, Nourelfath et Ait-Kadi, 2007, qui estiment qu'il est trop contraignant d'exiger des scans pour toutes les grumes pouvant être sciées, lorsque les horizons de planification font qu'une grande partie des grumes qui seront utilisées pour les réaliser n'ont pas encore été livrées aux scieries. Dans l'étude proposée, le simulateur de sciage est uniquement utilisé pour simuler le sciage d'un échantillon de grumes présentes dans une base de données historique. Ces résultats de simulation sont ensuite utilisés pour générer différents scénarios de production aléatoire et résoudre un problème d'optimisation par programmation stochastique. De même, Wery et al., 2018 proposent l'utilisation de techniques

de simulation-optimisation, Les procédures de simulation et d'optimisation sont alors réalisées itérativement. Les étapes de simulation génèrent des scénarios qui sont évalués et soumis à une procédure de recherche qui détermine les paramètres de la prochaine simulation. Le simulateur de sciage considère des grumes provenant d'une base de données non évolutive représentative de deux zones de récolte différentes. La simulation est particulièrement intéressante dans ce contexte car elle permet d'évaluer de nouvelles configurations de scierie en l'absence de données historiques.

Une troisième utilisation des outils de simulation est l'évaluation et la comparaison de différentes stratégies de planification. Par exemple, Mendoza et al., 1991 utilisent un modèle de simulation pour évaluer la faisabilité d'un plan produit par un modèle d'optimisation. Dumetz et al., 2015 considèrent un modèle de simulation à événements discrets pour comparer différentes stratégies de planification et de gestion des commandes. De même, Dumetz et al., 2021 l'utilisent pour comparer les stratégies de coordination entre les plans tactiques et opérationnels, et Cid Yáñez et al., 2009 proposent l'utilisation d'une plateforme de simulation multi-agents pour évaluer différents points de découplage entre flux tirés et poussés dans une chaîne logistique interne d'une scierie composée entre autres d'un agent de sciage, d'un agent de séchage et d'un agent de finition.

Un tel cadre de simulation multi-agents apparaît également dans de nombreux articles relatifs à l'industrie du sciage, même si l'on considère uniquement les problèmes de planification d'une seule scierie composée de plusieurs unités de production. La simulation multi-agents est originaire du champ de l'intelligence artificielle distribuée et permet donc, par nature, la mise en œuvre d'une stratégie de planification distribuée. Selon Gaudreault et al., 2010, ces technologies facilitent la création de plans plus détaillés que par l'approche centralisée, qui, en général, ne peut pas prendre en compte des détails opérationnels précis. Cela semble particulièrement intéressant pour une chaîne d'approvisionnement de scierie en raison de la nature divergente de plusieurs étapes de production qui compliquent la planification. La table 4.3 énumère les articles utilisant les technologies de simulation multi-agents pour étudier les problèmes de planification et de contrôle de la production dans l'industrie des produits forestiers et dans les scieries en particulier. Enfin, certains simulateurs de sciage intègrent un optimiseur pour émuler des lignes de sciage intégrant celles-ci. Un tel optimiseur peut, par exemple, décider du plan de coupe primaire ou secondaire à utiliser pour chaque grume.

Référence	**objectif du modèle de simulation**
Frayret et al., 2007	Conception et évaluation de stratégies de planification distribuées. Cet article se concentre sur l'introduction de la plate-forme de simulation développée par le Forac.
Cid et al., 2007	Évaluation et comparaison de stratégies de planification distribuées, avec différents points de découplage entre flux tirés et poussés.
Forget, D'Amours et Frayret, 2008	Cet article se concentre sur l'architecture générale des agents, mais propose un cas d'usage dans l'industrie forêt-bois.

Lemieux et al., 2009	Évaluation et comparaison de plusieurs stratégies de planification.
Cid Yáñez et al., 2009	Évaluation et comparaison de stratégies de planification distribuées, avec différents points de découplage entre flux tirés et poussés.
Forget et al., 2009	Évaluation et comparaison de mécanismes de coordination entre agents au niveau opérationnel.
Gaudreault, Frayret et Pesant, 2009	Coordination des opérations sur un horizon temporel court.
Gaudreault et al., 2010	Évaluation et comparaison de mécanismes de coordination.
Beaudoin, Frayret et LeBel, 2010	Coordination des opérations d'approvisionnement et de production entre plusieurs compagnies.
Santa-Eulalia et al., 2011	Évaluation et comparaison de plusieurs stratégies de planification.
Vahid, 2011	Optimisation de problèmes au niveau stratégique, comme le placement de nouvelles scieries.
Gaudreault et al., 2012	Évaluation et comparaison de mécanismes de coordination, basés sur un algorithme efficace de recherche de solutions à un problème d'optimisation.
Shahi et Pulkki, 2015	Optimisation des politiques d'inventaire pour l'aide à la décision tactique et opérationnelle.

Table 4.3 : Liste des articles rassemblés durant la revue de littérature traitant de l'utilisation de technologies multi-agents.

4.4 Ombres et jumeaux numériques en scierie

Bien qu'aucun des articles présentés dans la section précédente ne mentionne d'O&JN, ils soulignent l'existence de nombreux modèles numériques pouvant en constituer des composants élémentaires pour la prédiction, l'optimisation et plus généralement l'aide à la décision.

En raison de la nature très hétérogène de la matière première traitées par les scieries, de nombreuses technologies ont été mises au point par l'industrie pour mieux connaître leur nature tout au long du processus de transformation. Des méthodes ont été développées pour obtenir de telles informations dès l'arbre sur pied, avant la récolte. Nguyen et al., 2020, proposent, par exemple d'utiliser les scans Lidar, très étudiés dans le cadre de l'inventaire forestier, pour détecter et classer les défauts sur les troncs d'arbres sur pied. De même, l'industrie du sciage utilise depuis longtemps des scans externes et internes des grumes et des produits de sciage. Ces scans sont utilisés dans l'industrie à des fins diverses, notamment pour optimiser les processus, et il existe différentes technologies de scanner. En particulier, les scans laser de la forme extérieure des grumes sont utilisés depuis longtemps pour les calibrer et les classer, en mesurant leur longueur, leurs diamètres à chaque extrémité et leur volume. Des scanners à rayons X ont été développés pour permettre une détection plus poussée des défauts internes des grumes. Ces scans 3D peuvent prendre en compte les défauts internes des grumes ou uniquement sa forme. Ils sont utilisés dans l'industrie pour optimiser le tronçonnage des tiges en grumes ou la décomposition primaire des grumes en produits de sciage. De tels scans peuvent également être utilisés pour simuler le processus de sciage de manière non destructive et générer des données utiles à divers problèmes d'aide à la décision (Maness et Norton, 2002 ; Morneau-Pereira et al., 2014 ; Wery et al., 2014).

Tout comme les scans de la grume complète, les scans des produits semi-finis mesurés après leur décomposition primaire peuvent être utilisés pour l'optimisation du processus ou le contrôle de la qualité. Staudhammer, Maness et Kozak, 2007, en particulier, ont signalé l'utilisation de scans laser de la surface des produits de sciage pour le contrôle qualité.

Dans l'ensemble, les scans 3D, notamment ceux obtenus à partir de technologies à rayons X, peuvent être considérés comme suffisamment informatifs pour constituer les bases d'ombres numériques des grumes et des produits, à condition que ces représentations puissent suivre les bois sur plusieurs étapes de leur cycle de vie et être adaptées lorsque des étapes de transformation sont effectuées. Toutefois, cela soulève la question de la traçabilité des produits à travers plusieurs étapes de production divergentes. La traçabilité est un enjeu majeur dans l'industrie des produits forestiers pour limiter la perte d'informations qui a lieu à différentes étapes de la chaîne d'approvisionnement des produits forestiers, permettant ainsi de meilleures garanties sur l'origine et les caractéristiques des produits et une meilleure utilisation des ressources au niveau de la production en ligne (Jover et al., 2010). Différentes méthodes ont été étudiées à cette fin. En particulier, l'utilisation de scanners à rayons X semble prometteuse (Skog, Jacobsson, Lycken et al., 2017). En outre, les fabricants d'équipements proposent des technologies permettant la traçabilité des grumes aux bois de charpente, sur la base de scans à rayons X et de ce qu'ils appellent une empreinte digitale numérique (digital fingertip en anglais). Une autre méthode étudiée dans la littérature est le marquage des grumes à l'aide de marqueurs chimiques (Jover et al., 2010).

Par ailleurs, de nombreux modèles sont déjà capables de tirer parti de données industrielles telles que les scans des grumes pour déduire des informations sur les futures étapes de transformation. En particulier, certains fabricants de matériel de sciage fournissent des simulateurs qui intègrent

les même logiciels que ceux faisant fonctionner leurs machines. Ces simulateurs peuvent être reliés par les directeurs de scierie pour simuler l'ensemble de la chaîne de production et peuvent, par exemple, être utilisés pour évaluer les effets de changements dans la configuration de l'usine. Ils apparaissent donc comme des éléments importants d'un hypothétique O&JN d'usine, avec l'avantage d'être déjà présents, connus des gestionnaires de scierie, et littéralement utilisés par l'équipement à des fins d'optimisation locale. Il est intéressant de noter que certains de ces logiciels ont la capacité d'estimer grossièrement les résultats des futures étapes de production dans leur optimisation et peuvent donc déjà être considérés comme des ombres numériques de la ligne de production dans une certaine mesure. De même, il existe des simulateurs de sciage non liés à un équipement spécifique, tels que SAWSIM ou Optitek (Goulet, 2006). Optitek, par exemple, a été utilisé dans plusieurs études pour générer des données pour des problèmes d'optimisation. Parmi ces études, on peut citer Morneau-Pereira et al., 2014 ; Wery et al., 2018, qui mentionnent toutefois le long temps de calcul requis par de telles simulations. Par conséquent, Morneau-Pereira et al., 2014 effectuent des simulations pour un sous-ensemble de grumes seulement, tandis que Wery et al., 2018 proposent une méthode de simulation-optimisation pour réduire le nombre de simulations requises.

De nombreux modèles ont été proposés dans la littérature scientifique pour soutenir la prise de décision dans l'industrie du sciage, répondant à un large éventail de problèmes. En particulier, la recherche opérationnelle est largement utilisée par l'industrie des produits forestiers (Rönnqvist, 2003) et bénéficie de données plus précises que celles utilisées actuellement (Morin et al., 2020). De tels modèles sont capables de proposer des plans de production optimisés aux gestionnaires de scieries et répondent en particulier au problème d'optimisation du mélange de grumes à scier, et avec quel réglage d'équipement. Ceci est particulièrement important pour optimiser l'utilisation des matières premières et limiter la production de sous-produits de faible valeur. Cependant, il est important de noter qu'une formulation courante de ce problème nécessite, au minimum, que les grumes soient triées en classes basées sur des propriétés facilement mesurables mais ayant un fort impact sur le panier de produits comme leurs longueurs et diamètres. Bien que toutes les scieries ne mettent pas en œuvre l'étape requise pour utiliser de tels modèles de planification des opérations de sciage, d'autres modèles sont largement utilisés par les scieries canadiennes, par exemple pour planifier les opérations de séchage et de finition. Les simulateurs sont également largement utilisés pour soutenir la prise de décision aux niveaux stratégique et tactique plutôt qu'aux niveaux en ligne et opérationnel.

De même, ces modèles nécessitent diverses données, telles que les coûts d'exploitation, prix des produits, listes de commandes et rendement moyen des plans de coupes. Ces résultats, en particulier, ne sont pas toujours disponibles pour les nouveaux produits ou les nouvelles machines. Des modèles de simulation ont été proposés pour compléter ces données manquantes. Néanmoins, cela augmente le coût de calcul. De plus, l'hétérogénéité des tailles et organisation des scieries rend difficile la mise en œuvre de certains modèles pour des cas autres que celui pour lesquels ils ont été spécifiquement développés. Des cadres adaptatifs ont toutefois été proposés pour mettre en œuvre ces modèles au cas par cas (Kaltenbrunner, Huka et Gronalt, 2020). Les simulations doivent également être utilisées pour vérifier la faisabilité des plans proposés.

L'intérêt de rendre modulaire la structure des O&JN rend la littérature sur la simulation multi-agents particulièrement intéressante. Une scierie est composée de plusieurs unités de production distinctes. Chacune d'entre elles devrait naturellement être attribuée à son propre ensemble de modèles prédictifs potentiellement dépendants. Certains modèles peuvent avoir besoin des résultats des modèles des étapes de production précédentes comme entrées. Par conséquent, des

mécanismes sont nécessaires pour les coordonner à l'intérieur d'un cadre unique et optimiser le processus de production de manière globale, plutôt que de fournir un ensemble de solutions optimales seulement localement.

Ainsi, de nombreux éléments constitutifs des systèmes d'O&JN pour l'aide à la décision en scieries existent déjà, que ce soit dans la littérature scientifique ou dans l'industrie. On peut s'attendre à ce qu'un tel système limite la perte d'informations entre les différentes étapes de production de la scierie et améliore les plans de production à court et moyen termes en réduisant les incertitudes, notamment sur la matière première et la production attendue. Une meilleure coordination entre les unités de production peut également apporter d'énormes bénéfices, comme le montrent de nombreuses études basées sur des simulations multi-agents.

En outre, si aucun des articles examinés ne mentionne le concept d'O&JN, plusieurs autres articles traitent de leurs avantages dans le secteur forestier. Reitz, Schluse et Rossmann, 2019, par exemple, envisagent la mise en place d'un réseau O&JN pour coordonner et partager des informations entre les différents acteurs forestiers et leurs clients. Cette proposition vise à relever des défis spécifiques, tels que la faiblesse des connexions Internet dans les forêts et la réticence des acteurs à partager des informations. Kogler et Rauch, 2020 proposent un ensemble de modèles numériques, comme précurseur de O&JN, pour gérer le transport des forêts vers les scieries en tenant compte de l'incertitude liée aux événements naturels. Müller, Jaeger et Hanewinkel, 2019 présentent le concept de forêt numérique, c'est-à-dire une représentation numérique d'une forêt. Cette représentation rassemblerait des informations sur les conditions du sol, la topographie et les arbres individuels, en utilisant la télédétection, comme les technologies satellitaires, aéroportées ou LiDAR. Par conséquent, une forêt virtuelle est une ombre numérique de la forêt. La compléter par des capacités analytiques permettrait de créer un O&JN forestier. L'intégrer avec un O&JN de scierie, pourrait réaliser un outil puissant d'aide à la décision, permettant de coordonner, de simuler et d'optimiser des plans de production intégrés entre ces acteurs. Cela permet de limiter l'incertitude que subissent les scieries sur la matière première disponible en recueillant et en transmettant des informations très tôt dans la chaîne d'approvisionnement. Les technologies multi-agents joueront, là encore, un rôle clé pour assurer une coordination fluide et efficace entre ces différents O&JN. Des modèles ont été proposés pour gérer l'approvisionnement des scieries au niveau opérationnel ou en ligne et répondre aux événements inattendus (Amrouss et al., 2017). En analysant la qualité et les formes des grumes dès les opérations de récolte, voire avant, et en transférant ces informations aux scieries, les gestionnaires auraient de bien meilleures garanties sur la matière première transformée et réduiraient les incertitudes de production. Cela permettrait une meilleure maîtrise de la production et augmenterait la flexibilité des scieries.

4.5 Lien avec la stratégie de couplage proposée

Le temps en calcul important de certains simulateurs de sciage génériques a été mentionné par plusieurs chercheurs (Morneau-Pereira et al., 2014 ; Wery et al., 2014 ; Wery et al., 2018 ; Morin et al., 2020). Bien que les logiciels faisant fonctionner les lignes de sciage soient également proposés comme simulateurs par les fabricants, ils sont contraints d'effectuer une optimisation en temps réel, et peuvent ne pas trouver un véritable optimum. De plus, de nombreuses optimisations de ce type doivent être réalisées pour optimiser les plans de production. C'est le cas même en utilisant des modèles d'optimisation stochastique nécessitant seulement des résultats de simulations pour un sous-ensemble représentatif de grumes comme proposé par Zanjani, Nourelfath et

Ait-Kadi, 2013b, ou des méthodes de simulation-optimisation. Marier et al., 2014 mentionnent toutefois la difficulté, dans certains cas, de résoudre même un modèle déterministe simple de programmation mixte en nombres entiers assez rapidement pour un usage industriel au niveau opérationnel. De plus, alors que les paramètres de ces modèles peuvent être modifiés facilement pour répondre aux changements de la demande ou du mélange de matière première, leur structure globale reste fixe, ce qui peut avoir un impact négatif sur la capacité d'adaptation de l'O&JN et sa réponse aux changements de contexte et de processus. Proposer des modèles capables de réagir dans une certaine mesure à des changements inattendus, comme le font Amrouss et al., 2017 pour résoudre un problème de transport en temps réel dans le secteur forestier, est important. De même, certaines des technologies mentionnées restent hors de portée pour de nombreuses scieries.

Pour accélérer le processus de simulation, la construction de métamodèles de ces simulateurs de sciage semblent être une piste prometteuse. Morin et al., 2020, en particulier, proposent d'utiliser des algorithmes d'apprentissage automatique pour former des métamodèles de tels simulateurs. Ils proposent d'utiliser des algorithmes d'apprentissage automatique classiques et bien maîtrisés, tels que les forêts aléatoires ou les k-plus proches voisins, pour faire des prédictions basées sur des descripteurs métiers des grumes. Des études récentes ont prolongé ces travaux en explorant l'utilisation de divers autres algorithmes d'apprentissage automatique, tels qu'un kNN basé sur la dissimilarité ICP (Chabanet, El-Haouzi et Thomas, 2021 ; Chabanet et al., 2021) ou des réseaux de neurones profond (Martineau et al., 2021). Malgré ces travaux, il reste encore des recherches à faire sur leur structure, leur utilisation, leur contrôle et leurs limites. Les métamodèles de simulateurs de sciage sont, jusqu'à présent, spécifiques à une unique configuration de la scierie utilisés pour générer la base de données labélisées pour l'entraînement du modèle. Les processus de production qui se déroulent dans l'industrie du sciage sont cependant divers et varient en fonction des pays, des stratégies des entreprises et de la configuration des équipements. Il est donc nécessaire d'entraîner et de mettre en œuvre de nombreux métamodèles, ce qui induit un coûts en calcul et un temps de modélisation élevés. Un deuxième défi général est la présence habituelle de la dérive des concepts dans l'industrie.

Ainsi, le couplage en ligne de ces simulateurs de sciage et de leur métamodèle, pour la simulation des paniers de produits de tout ou partie des grumes livrées à la scierie avec une ou plusieurs configurations des lignes de sciage pourrait alors constituer un élément central d'une ombre numérique pour l'aide à la décision en scierie. Les informations générées ainsi pourraient alors être utilisées, par exemple, pour le tri des grumes dans le parc à grumes ou l'optimisation de plans de production.

Dans un tel système, le modèle de simulation est le modèle de l'unité de sciage d'une scierie, et les points de données transmis par le flux d'arrivée sont les scans des grumes nouvellement scannées à leur arrivée en scierie. Dans les travaux menés durant cette thèse, ces scans sont des nuages de points représentant la forme externe de la grume. Ils peuvent être utilisés directement par certains simulateurs de sciage, mais ne peuvent être utilisés tel quels par la plupart des modèles d'apprentissage automatique. Il est donc nécessaire de construire une représentation structurée de chaque scan. En reprenant les notations de la figure 3.1, les S_i sont les scans de chaque grume, et les $X_i = h(S_i)$ leur représentation structurée. Les y_i encodent les paniers de produits prédits par le simulateur de sciage à partir des scans S_i.

4.6 Conclusion

Ce chapitre présente une revue de littérature scientifique traitant de la planification et du contrôle de la production en scierie. Des problèmes d'aide à la décision ont été identifiés à tous les niveaux de planification, allant du niveau stratégique au niveau en ligne. Un problème particulièrement important, au niveau opérationnel, porte sur la décision du mix de grumes à scier, et avec quel plan de coupe. L'intégration horizontale et verticale de plans est également très étudiée.

Deux technologies très utilisées dans la littérature traitant de l'aide à la planification en scierie sont identifiées :

— les outils de recherche opérationnelle, en particulier la programmation linéaire et ses généralisations
— la simulation. En particulier, il existe, dans l'industrie et le monde académique, des simulateurs de sciage capable de prédire le panier de produits obtenu du sciage d'une grume particulière à partir d'informations sur sa forme et parfois ses défauts internes.

Il a, par ailleurs, été proposé de combiner ces différents outils, par exemple en utilisant des modèles de simulation pour générer les données requises par certain modèles d'optimisation qui bénéficient de données précises.

Ces résultats sont ensuite analysés sous l'angle des O&JN.

Bien que très peu d'articles traitent explicitement de l'usage des O&JN en scierie, le fait que les outils de simulations soient très présents dans cette industrie est un point encourageant pour le développement d'ombres numériques pour l'aide à la décision en scierie. Plusieurs auteurs mentionnent, cependant, le temps en calcul assez lent des simulateurs de sciage, rendant leur utilisation difficile pour l'aide à la décision opérationnelle.

Chapitre 5

Application de la méthode de couplage proposée au cas des scieries

5.1 Introduction

Le chapitre précédant a démontré l'intérêt des outils numériques pour l'aide à la décision dans la filière forêt-bois, en particulier les outils de simulation seuls ou combinés à des outils d'optimisation. La problématique des temps important associés à ces modèles de simulation a cependant été relevée. A ce titre, ce chapitre présente une application de la stratégie de couplage proposée chapitre 3 pour le développement d'une ombre numérique de scierie pour l'aide à la décision opérationnelle. Cette ombre numérique est constituée d'un simulateur de sciage et de son métamodèle, entraîné par apprentissage automatique. L'objectif de ces deux modèles est la prédiction des paniers de produits de sciage associés à des grumes de bois, au fur et à mesure qu'elles sont scannées. Les expériences présentées dans ce chapitre sont basées sur un jeu de données de résultats de simulation obtenu avec le logiciel Optitek, développé par FPInnovation [5]. Les scans utilisés pour ces simulations sont des scans de grumes réelles.

La suite de ce chapitre est organisée comme suit. La section 5.2 détaille le problème d'apprentissage auquel est entraîné le simulateur, puis la section 5.3 détaille le jeu de données utilisé tout au long de ce chapitre. Plusieurs représentations structurées des scans et algorithmes d'apprentissage sont comparées dans la section 5.4. La section 5.5 applique la stratégie de couplage au métamodèle considéré et analyse la sensibilité de la méthode à certains de ses paramètres. Enfin, la section 5.6 conclut ce chapitre.

5.2 Problème d'apprentissage

Les métamodèles entraînés dans ce chapitre sont des approximations d'un simulateur de sciage, paramétré pour une configuration d'une scierie. La tâche de prédiction à laquelle sont entraînés

5. `https://www.fpinnovations.ca/`, dernier accès Mars 2023

ces métamodèles est donc la prédiction du panier de produits $\hat{y}_i$ associé à une grume de scan S_i, à partir d'une représentation structurée de ce scan, $X_i = h(S_i)$. Plusieurs représentations structurées sont comparées dans ce chapitre.

Historiquement, ce problème d'apprentissage a été modélisé soit comme un problème de classification soit comme un problème de régression (Martineau et al., 2021). S'il est modélisé comme un problème de classification, l'objectif du modèle est la prédiction d'un panier de produits parmi une liste de paniers observés dans la base d'exemples d'entraînement. L'inconvénient est que plusieurs centaines de paniers différents peuvent exister dans ces jeux de données, certains extrêmement rares. De plus, il n'est pas garanti que tous les paniers possibles soient présents dans la base d'entraînement. Si le problème est, au contraire, modélisé comme un problème de régression, le panier de produits est modélisé comme un vecteur de dimension p, avec p le nombre de produits standards dans le catalogue de la scierie modélisée. La j^e composante du vecteur compte le nombre de produits du j^e type présents dans le panier de produits. Ainsi, n'importe quel panier imaginable peut théoriquement être prédit. Cela signifie aussi qu'il est possible de prédire des paniers de produits irréalistes en pratique. Les paniers prédits par des modèles de régression contiennent typiquement des nombres de produits non entiers, voir négatif dans certain cas. Ces prédictions ne sont donc utilisables que retraitées individuellement, ou agrégées par lot de billes.

Dans ce mémoire, nous nous concentrons sur la modélisation de ce problème d'apprentissage comme étant un problème de régression. Ce choix est fait car un grand nombre de paniers de produits uniques sont présents dans la base de données. Par ailleurs, beaucoup de ces paniers étant uniques, le jeu de données est fortement déséquilibré. Cela complique significativement l'utilisation d'algorithmes de classification.

5.3 Jeu de données

Le jeu de données utilisé dans cette section est fourni par le consortium Forac [6] et est originaire de l'industrie canadienne. Ce jeu de données comprend 2219 points de données. Chaque point de données contient des informations sur une grume différente. Les essences de ces grumes sont pin, sapin et épinette, trois essences résineuses très présentes dans les forêts canadiennes. Il peut être remarqué que les résineux constituent également une part très importante des volumes de sciage français (83% en 2017 [7]). Chaque point de données est, ainsi, constitué du scan 3D de la surface de la grume, de ses caractéristiques métiers, et du panier de produits obtenu par simulation avec le logiciel Optitek.

Un exemple de scan est présenté figure 5.1. Ces scans sont des listes de coordonnées cartésiennes de points, en 3D. Elles sont rangées dans une matrice $N_p \times 3$ avec N_p le nombre de points du scan. Ce nombre varie d'un scan à l'autre et dépend, en particulier, de la longueur de la grume considérée. Les points sont organisés en ellipsoïdes grossiers qui, ensemble, couvrent la surface de la grume.

En plus de ces scans, chaque grume est décrite par six caractéristiques métiers, couramment

6. `https://www.forac.ulaval.ca/en_bref/`, Dernier accès Novembre 2022
7. `http://chalayer-scierie.chez-alice.fr/pdf/2019/Dossier%20scierie%202025%20Bois%20int..pdf`, dernier accès Mars 2023

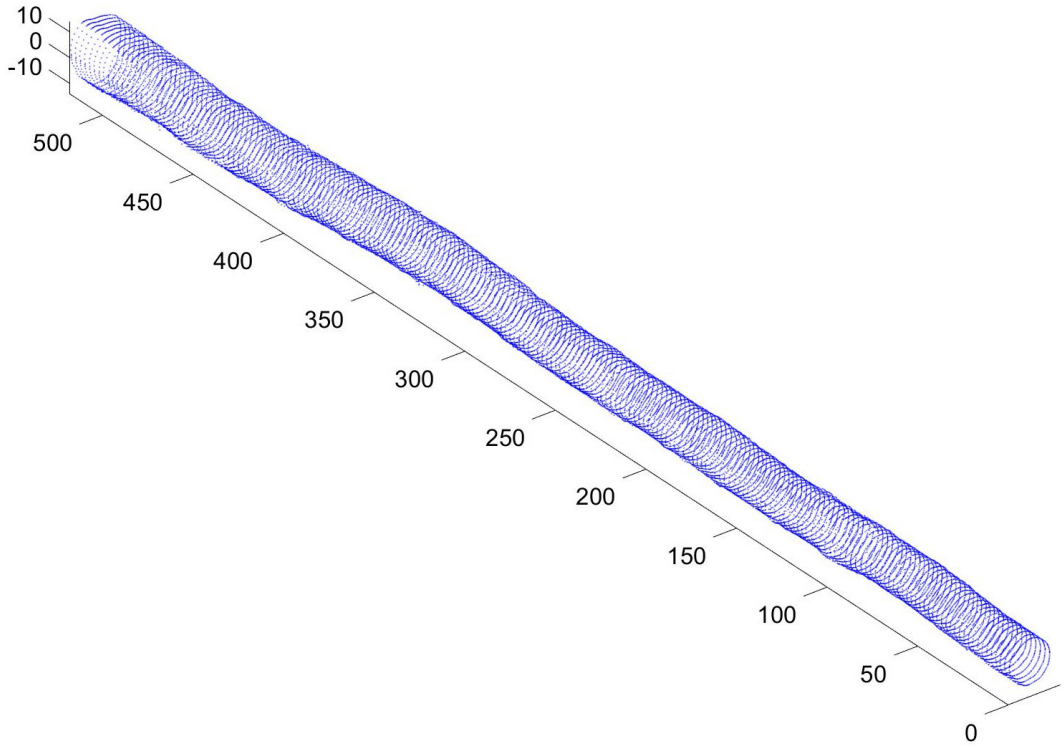

FIGURE 5.1 : Exemple de scan 3D d'une grume de bois

utilisées dans l'industrie pour décrire les grumes. Ces six caractéristiques sont la longueur, le volume, le diamètre à chaque extrémité, la courbure et la conicité (une mesure de la réduction du diamètre de la grume d'une extrémité à l'autre). Ces caractéristiques sont, en particulier, utilisées pour la prédiction de paniers de produits de sciage par MORIN et al., 2015 et MORIN et al., 2020.

La scierie simulée par le logiciel Optitek est capable de produire 47 types de produits standards. Ces produits sont caractérisés par leurs longueur, largeur et épaisseur. La qualité du produit de sciage, pouvant varier entre produits de même dimensions, n'est pas considérée ici pour limiter le nombre, déjà grand, de types de produits à prédire. Au total, 870 paniers différents sont observés dans cette base de données, dont 614 sont uniques.

5.4 Choix du métamodèle

5.4.1 Apprentissage par fonction de proximité :

Cette section n'est pas centrée sur les O&JN mais présente les outils que seront utilisés pour pouvoir appliquer l'approche proposée au chapitre 3 au cas où les données d'apprentissage sont des nuages de points 3D.

Les nuages de points 3D sont une source de données de plus en plus importante avec la démocratisation des méthodes d'acquisition tels que les caméras lidar. Ce type de données se retrouve dans des secteurs d'activité variés tels que le bâtiment (Gama, Sebastiao et Rodrigues, 2013), la médecine (Yu et al., 2021) ou les scieries (Selma et al., 2018). Les problèmes levés par ce type de données en apprentissage automatique sont, par exemple des problèmes de segmentation ou de classification (Qi et al., 2017). L'application qui est présentée dans ce chapitre se base notamment sur un jeu de données constitué de scans de billons de bois, c'est-à-dire des nuages de points représentant la forme des billons. L'un des freins à l'utilisation à ce type de données de méthodes usuelles d'apprentissage automatique est leur manque de structure. Les points contenus dans ces nuages ne sont pas nécessairement ordonnés, et leur nombre peut varier d'un nuage à un autre.

Un pan important de la littérature se rapportant à l'apprentissage automatique pour le traitement de nuages de points est basé sur l'utilisation d'architectures spécifiques de réseaux de neurones profonds capables de traiter ces données malgré leur manque de structure. L'architecture pointnet (Qi et al., 2017) intègre par exemple un certain nombre de couches réalisant des opérations point par point, sans tenir compte de leur ordre ou nombre, suivies de couches de max pooling permettant de ne sélectionner qu'un nombre fixe de points pour la classification du nuage. L'inconvénient principal des modèles d'apprentissage profond est cependant leur coût très important à la fois en ressources computationnelles et en quantité de données nécessaires à leur entraînement. Pour cette raison, ces méthodes n'ont pas été considérées dans ce travail de thèse.

L'approche considérée ici se base sur l'apprentissage par fonction de proximité, et plus particulièrement par fonctions de dissimilarités. Une fonction de dissimilarité est une fonction $d : \mathbf{F} \times \mathbf{F} \to \mathbf{R}$ qui, intuitivement, estime une ressemblance entre deux objets de l'espace d'entrée $\mathbf{F}$, dans notre cas, deux nuages de points. Ces fonctions de proximité sont donc similaires à des fonctions distances, mais n'en possède pas forcément les caractéristiques. Une fonction de dissimilarité peut ne pas être symétrique, ou ne pas respecter l'identité triangulaire.

De telles fonctions de proximité apparaissent naturellement dans de nombreux problèmes pour lesquels il est difficile de décrire certains objets par un nombre fixe de caractéristiques. Nous pourrons par exemple citer la dissimilarité Dynamic Time Warping (Müller, 2007) pour comparer des séries temporelles, l'alignement de Smith-Waterman (Smith, Waterman et al., 1981) pour comparer des séquences génétiques ou la similarité de Jaccard (Luo, Li et Chung, 2009) pour comparer des documents textes.

Quelques méthodes d'apprentissage par fonction de proximité

De nombreuses méthodes ont été proposées dans la littérature scientifique pour permettre l'apprentissage de modèles de prédiction à partir de données sous forme d'évaluations de fonctions de proximités sur des paires d'objets.

Beaucoup se basent notamment sur la modification de la matrice de similarité M qui regroupe les mesures de (dis)similarité entre toutes les paires d'objets présentes dans la base de données d'entraînement du modèle d'apprentissage. Certains algorithmes d'apprentissage automatique, comme les SVM, peuvent en effet apprendre de telles matrices à condition que celles-ci soient semi-définies positives. Il est donc nécessaire de modifier la matrice M pour la rendre semi-définie positive. Une revue de ces méthodes est présentée dans Schleif et Tino, 2015. Elles se basent souvent sur la modification des valeurs propres de M. Ces méthodes peuvent cependant

présenter un coût algorithmique important, jusqu'à Ω^3 pour l'entraînement, avec Ω la taille de la base d'entraînement, et l'évaluation de nouveaux points de données non présents dans la base d'entraînement peut également s'avérer complexe.

Une deuxième famille de méthodes regroupe les algorithmes capables d'apprendre directement de matrices de (dis)similarités non métriques. C'est par exemple le cas des méthodes basées sur les espaces de Krein à noyaux reproduisant, qui généralisent la théorie des espaces de Hilbert à noyaux reproduisant, sur laquelle s'appuient notamment les modèles SVM (Loghin et Ismonov, 2022). Il est intéressant de noter que toute matrice de similarité de dimension finie peut être plongée dans un espace de Krein. Cependant, toute fonction de similarité, même symétrique, n'admet pas de représentation générale sous forme d'un produit scalaire tel que défini dans la théorie des espaces de Krein.

Enfin, une troisième méthode est le plongement des points de données dans un sous-espace de dissimilarité (Duin et Pękalska, 2009). L'objectif de cette méthode est d'utiliser des mesures de dissimilarités pour construire un espace euclidien de dimension m, fixé par l'utilisateur. La première étape consiste en la sélection d'un sous-ensemble de m points représentatifs $\mathbf{R} = S_1, ..., S_m \in \mathbf{F}^m$, appelés prototypes. Considérant la fonction de proximité $d : \mathbf{F} \times \mathbf{F} \to \mathbf{R}$, tout point S de $\mathbf{F}$ peut être plongé dans un espace euclidien défini par la transformation :

$$X = h(S) = (d(S, S_1), ..., d(S, S_m)) . \tag{5.1}$$

Une manière d'interpréter ce plongement est qu'il construit un vecteur de caractéristiques, chaque caractéristique étant définie comme la dissimilarité à un prototype. Cette méthode présente de nombreux avantages. D'abord, elle permet l'utilisation de tout modèle d'apprentissage automatique nécessitant un vecteur de caractéristiques en entrée, comme les réseaux de neurones ou les forêts aléatoires. Ces modèles sont bien connus et maîtrisés, et il en existe des implémentations efficientes. De plus, la prédiction de nouveaux points de données ne requiert le calcul que de m dissimilarités. Ces calculs pouvant être relativement coûteux, cela laisse à l'utilisateur le contrôle du nombre de calculs de dissimilarités. Cependant, les modèles d'apprentissage basés sur ces plongements ne considèrent plus la nature des caractéristiques en tant que (dis)similarités. De plus cette méthode pose le problème de sélection de caractéristiques, sous la forme de la sélection initiale des prototypes.

Il a été montré empiriquement par Pękalska, Duin et Paclík, 2006 qu'une sélection aléatoire des prototypes dans la base de données d'entraînement peut donner de bons résultats, même comparativement à des méthodes systématiques, et que le choix du modèle de prédiction présente un impact bien plus important sur la précision des prédictions. Cependant des auteurs comme Kar et Jain, 2011 défendent l'utilisation d'heuristiques assurant que les prototypes soient aussi dissimilaires les uns des autres que possible pour réduire le nombre de prototypes nécessaires à l'apprentissage. Ils introduisent notamment une heuristique gloutonne, nommée Dselect, qui sélectionne itérativement le point de la base de données qui maximise la dissimilarité moyenne à tous les prototypes déjà sélectionnés.

Dissimilarités pour les nuages de points 3D

Les fonctions de dissimilarité entre nuages de points 3D peuvent présenter plusieurs invariants. Ces invariants peuvent être désirables ou non suivant le problème considéré :

— Invariance par rotation : la dissimilarité ne dépend pas de l'orientation des nuages de points.

— Invariance par translation : la dissimilarité ne dépend pas de la position de l'origine du repère servant à donner les coordonnées des points.

— Invariance par passage à l'échelle : La dissimilarité ne dépend pas de la taille des nuages de points comparés, uniquement de leur forme.

— Symétrie : étant donné deux nuages x_1 et x_2, la dissimilarité de x_1 par rapport à x_2 $d(x_1, x_2)$ est la même que la dissimilarité de x_2 par rapport à x_1 $d(x_2, x_1)$.

De plus certaines dissimilarités comparent directement les nuages de points tandis que d'autre utilisent une représentation intermédiaire. Cette deuxième approche peut notamment faciliter le stockage des données puisque seul un résumé du nuage de points doit être gardé. Trois dissimilarités seront comparées dans l'application présentée dans ce chapitre : La dissimilarité Itérative Closest Point (ICP), la dissimilarité Ensemble of Shape Function (ESF) et la dissimilarité multiview 2D projection (M2DP).

La dissimilarité ICP est une conséquence de l'algorithme Itérative Closest Point pour l'alignement de nuages de points. Plus précisément, la version utilisée est l'algorithme ICP point to point publié dans Besl et McKay, 1992. L'objectif premier de cet algorithme est l'alignement de deux nuages de points pour qu'ils soient aussi proches l'un de l'autre que possible. Les deux nuages ne jouent pas des rôles symétriques. Le premier, nommé la cible, reste fixe tout au long du déroulé de l'algorithme. Le second, appelé source, se voit appliquer une succession de translations et de rotations pour l'aligner sur la source.

L'algorithme ICP est itératif. A chaque itération, il cherche, pour chaque point $s_{s,i}$ dans la source S_s son plus proche voisin $s_{c,i}$ dans la cible S_c. Un même point de la cible peut être sélectionné plusieurs fois ou jamais. L'algorithme détermine alors un couple translation rotation (T, R) qui minimise la dissimilarité

$$d(RS_s + T, S_c) = \sum_i (Rs_{s,i} + T - s_{c,i})^2 \tag{5.2}$$

Ce problème d'optimisation admet une solution analytique. Le nuage est alors transformé et de nouveaux couples (s_i, c_i) obtenus, ce qui donne lieu à un nouveau problème de minimisation et ainsi de suite. Ces opérations sont répétées jusqu'à ce qu'un critère de fin soit atteint, par exemple qu'une certaine précision soit atteinte ou qu'un nombre maximum d'itérations aient été réalisées. Un pseudo code de cet algorithme est présenté algorithme 1. Il peut être montré que la quantité $d(RS + T, C)$ décroît à chaque itération et converge donc vers un minimum local. C'est la valeur de $d(RS + T, C)$ qui est alors considérée comme la dissimilarité ICP : d_{ICP}. Puisque cet algorithme ne converge que vers un minimum local, cependant, cette dissimilarité n'est pas invariante par translation ni par rotation. Elle n'est pas non plus invariante par passage à l'échelle, ni symétrique.

La complexité de l'algorithme ICP est dépendante du nombre de points dans les deux nuages comparés. En particulier, la construction naïve des paires (s_i, c_i) à chaque itération a un coût en $O(N_s.N_c)$ avec N_s le nombre de points dans le nuage source et N_c le nombre de points dans le nuage cible. Il est cependant possible d'accélérer cette recherche en utilisant des structures de données spécialisées, comme les arbres kd (Wang et al., 2015). Dans ce cas, la complexité est en $O(N_s.ln(N_c))$ en moyenne.

Cette dissimilarité a été utilisée pour la comparaison de scans de billons de bois par Selma et al., 2018.

Algorithm 1 Algorithme ICP point to point

Input Source S et cible C
Output rotation R et translation T.

R ← matrice rotation identité
T ← vecteur translation 0
while condition d'arrêt non atteinte **do**
  crée les paires (s_i, t_i) entre chaque point de S et son plus proche voisin dans C
  Trouve R_0, V_0 qui minimise $d(RS + V, T)$
  S ← $R_0 S + V_0$
  $R \leftarrow R_0 * R$
  $V \leftarrow R_0 * V + V_0$
end while

Contrairement à la dissimilarité ICP qui mesure directement la dissimilarité entre deux nuages de points, la dissimilarité ESF, d_{ESF}, construit d'abord une représentation intermédiaire des nuages et calcule une dissimilarité entre ces deux représentations. L'avantage de cette méthode est que la représentation de chaque nuage n'a à être calculée qu'une seule fois et peut être stockée telle quelle. Le calcul de dissimilarité entre ces deux représentations est ensuite très rapide.

Cette dissimilarité a été introduite par Osada et al., 2001. Chaque nuage de points est d'abord représenté par une collection de cinq histogrammes. Ces histogrammes présentent les distributions de valeurs obtenues en évaluant des fonctions, dites fonctions de formes, sur des petits groupes de points sélectionnés au hasard dans le nuage. Ces fonctions sont :

— $\mathbf{F_1}$ La distance entre le centroïde du nuage et un point pris au hasard.
— $\mathbf{F_2}$ La distance entre deux points pris au hasard.
— $\mathbf{F_3}$ La racine carrée de l'aire du triangle formé par trois points pris au hasard.
— $\mathbf{F_4}$ La racine cubique du volume du tétraèdre formé par quatre points pris au hasard.
— $\mathbf{F_5}$ L'angle formé par trois points pris au hasard.

La dissimilarité ESF est alors la somme des distances L_1 entre les histogrammes de chaque nuage de points :

$$d_{ESF}(x_i, x_j) = \sum_{k=1}^{5} ||F_{ki} - F_{kj}||_1 \,, \tag{5.3}$$

ou F_{ki} est le k^e histogramme du nuage x_i. L'avantage de la distance L_1 est d'être invariante par dilatation des histogrammes, c'est à dire indépendante de l'unité choisie, en particulier pour la mesure de l'angle.

Contrairement à la dissimilarité ICP, la complexité du calcul de la dissimilarité ESF ne dépend pas du nombre de points dans les nuages, mais plutôt du nombre de tirages effectués pour construire les histogrammes. De plus elle est symétrique, et invariante par rotation et translation puisque les fonctions de forme le sont. En revanche, elle n'est pas invariante par passage à l'échelle.

M2DP (He, Wang et Zhang, 2016) a été introduite dans le contexte de la cartographie et localisation simultanée par des robots mobiles. Comme l'algorithme ESF, il utilise une représentation intermédiaire du nuage de points, basée cette fois sur des projections dans plusieurs plans 2D. Cependant, cette représentation n'est pas naturellement invariante par translation et rotation. Le nuage est donc prétraité, de sorte que le centroïde du nuage soit placé à l'origine. Une Analyse en Composantes Principales (ACP) est également réalisée et les axes du nuage alignés suivant les axes principaux obtenus.

Dans un premier temps, le nuage est projeté suivant pq plans, p et q deux entiers. Chaque projection produit un vecteur de $\beta\gamma$ caractéristiques, β et γ deux entiers. Ces vecteurs sont concaténés en une matrice de dimension $pq \times \beta\gamma$. La version originale de cet algorithme résume encore cette matrice en ne gardant que les premiers vecteurs propres à gauche et à droite. Cette étape a pour conséquence de rendre ces caractéristiques indépendantes du nombre de points dans le nuage, et invariant par passage à l'échelle. Cette propriété n'étant pas souhaitable dans l'application développée dans cette thèse, cette étape n'est pas réalisée. La dissimilarité entre deux nuages de points S_i et S_j, représentés par les matrices M_i et M_j est alors prise comme la norme de Frobenius de la différence entre ces matrices :

$$d_{M2DP}(x_i, x_j) = \sum_{k,r} (M_i - M_j)^2_{kr} \tag{5.4}$$

Le vecteur décrivant chaque projection est construit de la façon suivante. Le plan de projection est d'abord séparé en β cercles concentriques, centrés sur la projection du centroïde du nuage. Chaque cercle est alors décomposé en γ sections, menant à $\beta\gamma$ cases. Le nombre de points dans chaque case est alors compté, menant au vecteur de lt caractéristiques. Un pseudocode de cet algorithme est présenté algorithme 2.

5.4.2 Méthodes comparées

Deux types de modèles d'apprentissage automatique croisés avec les quatre représentations structurées des scans de grumes présentés précédemment sont comparés pour la prédiction des paniers de produits de grumes.

La première représentation structurée est basée sur les six caractéristiques métiers des grumes utilisées par Morin et al., 2015 et Morin et al., 2020. Ces caractéristiques métiers sont comparées à 3 ensembles de caractéristiques, basées sur les dissimilarités entre un scan et un ensemble de prototypes, suivant la méthode de plongement dans un sous-espace de dissimilarité. Les trois dissimilarités utilisées sont la dissimilarité ICP, la dissimilarité ESF et la dissimilarité M2DP

Algorithm 2 Algorithme M2DP, d'après He, Wang et Zhang, 2016

Input Nuage de points x, nombre de cercles β, nombre de sections γ, nombre de plans de projection suivant l'azimut p, nombre de plans de projection suivant l'élévation q.

Output Matrice M de dimension $pq \times \beta\gamma$.

$x \leftarrow x - \bar{x}$, place le centroïde du nuage à l'origine
Calcul de l'ACP. Aligne l'axe des x avec la première composante principale. Aligne l'axe des y avec la seconde composante principale.
$x \leftarrow \mathbf{0}$ la matrice nulle $pq \times \beta\gamma$
for $k = 0$ to $p - 1$ **do**
  $\theta \leftarrow \frac{k\pi}{p}$
  for $r = 0$ to $q - 1$ **do**
    $\phi \leftarrow \frac{r\pi}{2q}$
    $m \leftarrow (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\phi)$ vecteur normal au plan de projection.
    $x \leftarrow x - (x^T.m)m$ projette le nuage de points sur le plan de normale m.
    Décompose le plan de projection en $\beta\gamma$ cases
    compte le nombre de points dans chaque case et complète la ligne $k + r$ de M
  end for
end for

présentées dans ce chapitre. La sélection des prototypes est réalisée avec l'heuristique CoreSelect. Cette heuristique est une variante de la méthode dselect proposée par Kar et Jain, 2011. La description et la motivation de cette heuristique est détaillée annexe B. Dans un premier ensemble d'expériences, le nombre de prototypes sélectionnés est fixé à 50. Ce paramètre est ensuite varié pour étudier son impact sur la performance des modèles d'apprentissage.

Les deux modèles d'apprentissage comparés sont des modèles de forêts aléatoires, déjà présentés, et des réseaux de neurones. Les modèles de forêts aléatoires sont testés car ce sont les modèles donnant les meilleurs résultats en prédiction dans Morin et al., 2015 et Morin et al., 2020. Les réseaux de neurones sont évalués pour leur popularité dans la littérature scientifique et leur capacité à approximer des fonctions complexes à un quelconque niveau de précision (Cybenko, 1989 ; Lu et al., 2017). Considérant la faible taille des bases de données d'entraînement considérées ici, nous nous restreignons, cependant, à des architectures perceptron à une seule couche cachée, avec un faible nombre de neurones et donc de paramètres à estimer.

Une comparaison de plusieurs architectures a été menée dans Chabanet, Thomas et El-Haouzi, 2021. Nous utilisons ici l'architecture ayant démontré les meilleurs résultats dans cette étude. Il s'agit d'une architecture Multi-Input Single-Output (MISO), contenant une seule couche cachée. En pratique, cela signifie qu'un réseau de neurone est entraîné indépendamment pour la prédiction de chaque type de produit de sciage. La fonction d'activation de la couche cachée est une fonction tangente hyperbolique. Une fonction sigmoide est également utilisée sur la dernière couche du réseau, pour contraindre (après transformation affine) la prédiction dans l'intervalle compris entre la plus faible et la plus haute valeurs observées sur la base de donnée d'entraînement. Pour leur entraînement, les poids de ces réseaux ont été initialisés avec l'algorithme Nguyen-Widrow (Nguyen et Widrow, 1990) pour accélérer la convergence des poids durant l'entraînement. L'entraînement des poids est réalisé avec l'algorithme Levenberg-Marquardt. L'implémentation de cet algorithme a été réalisée en python, basée sur la procédure décrite dans la documentation

Matlab [8]. En particulier, cette implémentation intègre plusieurs conditions d'arrêt permettant de limiter le surapprentissage du réseau. Plus précisément, l'entraînement est stoppé lorsque l'erreur moyenne de prédiction évaluée sur une base de validation ne décroît pas pour 6 itérations successives. Cette méthode requiert, cependant, l'utilisation d'une base de validation labélisée en plus de la base d'entraînement.

Le jeu de données a, ainsi, été séparé trente fois au hasard en une base d'entraînement, une base de validation et une base de test. Les bases d'entraînement et de validation contiennent 500 exemples d'entraînement chacune. La base de test contient les 1219 exemples restant. L'estimation des performances de chaque couple représentation/modèle d'apprentissage est réalisée par validation croisée Monte-Carlo (Arlot, 2018). Pour chaque séparation du jeu de données, les modèles d'apprentissage automatique sont entraînés sur la base d'entraînement. La base de validation est, en particulier, utilisée pour optimiser les hyper-paramètres de chaque modèle. Leurs erreurs quadratiques moyennes (Mean Square Error, MSE en anglais) sont ensuite évaluées sur leur base d'évaluation respective. Les MSE sont ensuite moyennées sur toutes les séparations du jeu de données. Cela permet de limiter la variabilité induite par la séparation du jeu de données en bases d'entraînement, de validation et de test.

En plus d'être utilisée pour limiter le surapprentissage des modèles MLP, les bases de validation sont systématiquement utilisées pour la sélection d'hyperparamètres. Pour les MLP en particuliers, dix MLP ayant des initialisations aléatoires différentes sont entraînés pour trois valeurs du nombre de neurones dans la couche cachée. Ces trois valeurs sont 2, 5 et 10. Le modèle MLP avec la plus faible MSE sur la base de validation est sélectionné. Il est ensuite réévalué sur la base de test. De même, plusieurs forêts aléatoires sont évaluées pour un nombre croissant B d'arbres dans la forêt prenant successivement les valeurs 100, 250 et 500. Un deuxième paramètre important pour les modèles de forêt aléatoire est le nombre de caractéristiques considérées lors des recherches des meilleures coupes pendant la croissance des arbres. Il correspond au paramètre *max_feature* de l'implémentation de la bibliothèque python sklearn. Ce paramètre est exprimé en fraction du nombre total de caractéristiques et est sélectionné pour chaque division de la base de donnée dans la liste $[0.1, 0.5, 1.0]$. Ce paramètre a un impact sur la corrélation des arbres entre eux. L'erreur quadratique d'un modèle f entraîné sur un jeu de données D pour une entrée x est, en effet, un compromis entre le biais de l'erreur $\mathbf{E}_D(f(x) - y)$ et sa variance d'échantillonnage $\mathbf{Var}_D(f(x))$. Une faible valeur de *max_feature* mène à des arbres moins corrélés ce qui réduit la variance. Cependant, puisque dans de nombreux cas pratiques certaines caractéristiques sont plus informatives que d'autres, baisser ce paramètre peut également augmenter le biais de l'erreur.

5.4.3 Scores d'évaluations

Dans cette section, la qualité d'une prédiction sera, comme dans le chapitre 3, évaluée principalement par son erreur quadratique. Cependant, plusieurs autres scores ont été introduits par divers auteurs pour évaluer spécifiquement la qualité de prédictions de métamodèles de simulateurs de sciage. En effet, lorsque le problème de prédiction du panier de produits est considéré comme un problème de classification, le score 0-1 classique ne prend pas en considération le fait que le coût d'une mauvaise classification change suivant quelle paire panier réel/panier prédit est considérée. De même, lorsque, comme c'est la cas dans ce mémoire de thèse, le problème de prédiction du panier de produits est considéré comme un problème de régression, l'erreur quadratique a le

8. `https://fr.mathworks.com/help/deeplearning/ref/trainlm.html`

désavantage d'être difficile à comprendre et interpréter par les experts métiers.

Pour cette raison, le score de prédiction, s^{pre}, le score de production, s^{pro}, et le score de prédiction-production, $s^{pre\times pro}$, ont été introduits par Morin et al., 2015. De même, les scores précision, rappel et F_1, très utilisés en classification, ont été adaptés à ce problème de prédiction (Martineau et al., 2021).

Introduisons d'abord les scores de prédiction, production, et prédiction-production. Ces scores sont systématiquement calculés entre un panier réel ou simulé, y_i, et une prédiction réalisée sur la même grume de bois, $\hat{y}$. Un score est donc associé à chaque prédiction. Il est également nécessaire de préciser que les vecteurs encodant les paniers de produits sont creux. Un seul panier contient peu de types de produits différents. Ainsi, la majorité des éléments de chaque vecteur sont égaux à 0. Une conséquence sur les scores de prédiction, production, et prédiction-production est que laisser toutes les paires $(0, 0)$ de quantités de produits réelle et prédite dans les calculs s^{pre}, s^{pro}, et $s^{pre\times pro}$ biaise ces scores vers le haut et les rend excessivement optimistes. Pour cette raison, toutes ces paires $(0, 0)$ réelles/prédites sont filtrées avant le calcul des scores. Notons $\tilde{p}$ la longueur des vecteurs dont ont été enlevées toutes ces paires. Le score de prédiction est définit comme :

$$s^{pre}(y_i, \hat{y}_i) = \frac{1}{\tilde{p}} \sum_{j=1}^{\tilde{p}} \min(1, \frac{\hat{y}_{ij}}{\max(\epsilon, y_{ij})}) \,, \tag{5.5}$$

avec y_{ij} et $\hat{y}_{ij}$ les j^e éléments des paniers réel et prédit respectivement. ϵ est une petite constante introduite pour éviter la division par 0. Elle est, ici, fixée à 0.00000001. Le score de prédiction est, ainsi, la moyenne par type de produits de la proportion de produits réels effectivement prédite.

De même, le score de production est défini comme :

$$s^{pro}(y, \hat{y}) = \frac{1}{\tilde{p}} \sum_{i=1}^{\tilde{p}} \min(1, \frac{y_i}{\max(\epsilon, \hat{y}_i)}) \,, \tag{5.6}$$

et peut être interprété comme la moyenne par type de produits de la proportion de produits prédit effectivement produite.

Une limite de ces scores considérés individuellement est, cependant, que toujours prédire un panier de produits vide induit un score de production maximal. De même, prédire un panier de produits avec une grande quantité de chaque type de produits maximise le score de prédiction. Pour équilibrer ces deux scores, le score de prédiction-production a donc été introduit comme le produit du score de prédiction et du score de production :

$$s^{pre\times pro} = s^{pre} \times s^{pro} \,. \tag{5.7}$$

Les scores précision, rappel et F_1, sont, quand à eux, basés sur une redéfinition des nombres de vrai positifs (en anglais True Positives, TP), faux positifs (en anglais False Positives FP) et faux négatifs (en anglais False Negatives FN) :

— TP est la quantité de produits présents à la fois dans le panier réel et dans le panier prédit, c'est à dire : $TP(y_i, \hat{y}_i) = \sum_{j=1}^{p} \min(y_{ij}, \hat{y}_{ij})$.
— FP est la quantité de produits présents dans le panier prédit mais pas dans le panier réel, c'est à dire : $FP(y_i, \hat{y}_i) = \sum_{j=1}^{p} \max(\hat{y}_{ij} - y_{ij}, 0)$.
— FN est la quantité de produits présent dans le panier réel mais pas dans le panier prédit, c'est à dire : $FN(y, \hat{y}) = \sum_{i=1}^{p} \max(y_i - \hat{y}_i, 0)$.

La précision, le rappel et le score F_1 sont définis avec les formules usuelles :

$$precision(y, \hat{y}) = \frac{TP(y, \hat{y})}{TP(y, \hat{y}) + FP(y, \hat{y})}\,, \tag{5.8}$$

$$recall(y, \hat{y}) = \frac{TP(y, \hat{y})}{TP(y, \hat{y}) + FN(y, \hat{y})}\,, \tag{5.9}$$

and

$$F_1(y, \hat{y}) = 2\frac{precision(y, \hat{y}) recall(y, \hat{y})}{precision(y, \hat{y}) + recall(y, \hat{y})}\,. \tag{5.10}$$

Ces scores sont, comme s^{pre}, s^{pro}, et $s^{pre\times pro}$ calculés indépendamment pour chaque prédiction. Une valeur est donc calculée pour chaque grume. La quantité d'intérêt pour l'évaluation d'un modèle prédictif est, cependant, la moyenne de ces scores sur un grand nombre de prédictions. Dans la suite de ce mémoire, les notations $s^{pre}, s^{pro}, s^{pre\times pro}, precision, recall$ and F_1 feront donc référence aux moyennes observées sur les bases de données de test plutôt qu'aux mesures individuelles. De plus, ces scores ayant été conçus pour des prédictions à valeur entières, et le score de prédiction-production étant particulièrement sensible à la sparsité des prédictions, ces scores, et ces scores uniquement, sont calculés sur les prédictions arrondies à l'entier le plus proche.

5.4.4 Résultats

Les moyennes et écart-types sur les 30 répétitions de la base de données des différents scores évalués sont présentés table 5.1. Les meilleurs scores moyens sont systématiquement obtenus pour le modèle de forêt aléatoire après plongement des grumes dans un sous-espace de dissimilarité ICP. Le deuxième meilleur modèle, quel que soit le score considéré, est le modèle de forêt aléatoire utilisant les caractéristiques métiers. En particulier, la différence entre les RMSE obtenus pour ces deux meilleurs modèles peut être considérée statistiquement significative, par rapport aux répétions des expériences. Un test de Student comparant la moyenne des 30 RMSE obtenues avec les caractéristiques métiers avec les 30 RMSE obtenues avec les caractéristiques ICP donne, en effet, une p-value de 7.9×10^{-11}. C'est donc ce modèle et cette représentation qui seront utilisés dans la suite des expériences présentées dans ce mémoire. Il doit être souligné que ce test, ainsi que tous les tests de Students réalisés par la suite, ne peuvent être considérés que conditionnellement au jeu de données global. Ils considèrent, notamment, la variabilité induite par le processus expérimental, comme la séparation en base d'entraînement et d'évaluation, mais ne considèrent pas la variabilité sous-jacente à la production du jeu de données.

Les scores prédiction-production et F_1 les plus faibles, ainsi que les RMSE les plus hautes, sont, au contraire, obtenus avec les caractéristiques M2DP quel que soit le modèle d'apprentissage utilisé. La différence avec le modèle MLP utilisant les caractéristiques ESF est, en particulier, significative suivant un test de Student similaires à ceux réalisés aux préalables. Deux hypothèses peuvent être faites pour expliquer ce fait, en lien avec l'algorithme M2DP. La première est que le découpage des plans de projections en voxels dans l'algorithme M2DP entraîne une perte trop grande d'information fine sur la forme des grumes. La deuxième est que le découpage des voxels en portion de cercles n'est pas adapté à la géométrie des grumes, projetée en 2D.

Concernant les paramètres de ce modèle, le nombre d'arbres sélectionnés le plus fréquemment sur les bases de validations est 500, qui est sélectionné 14 fois sur 30. Il est à noter qu'un nombre d'arbres grand permet également de réduire le biais de l'estimateur de variance d'échantillonnage définit équation 3.22. De façon plus surprenante, la valeur du paramètre *max_feature* sélectionnée le plus fréquemment est 0.1, sélectionné lors de 21 répétitions de l'expérience. Un comportement similaire est observé sur toutes les forêts aléatoires utilisant des caractéristiques basées sur des mesures de dissimilarité, mais pas sur celles utilisant les caractéristiques métiers qui favorisent plutôt une grande valeur de ce paramètre, $max\_feature = 1.0$ étant sélectionné sur 26 des 30 expériences. Ce choix du paramètre peut s'expliquer par le fait que, par construction, aucun sous-échantillon aléatoire des caractéristiques du sous-espace de dissimilarité n'a de raison de mener à des coupes plus performantes. Contrairement aux features métiers, ils contiennent tous une information de nature similaire. Cependant, une faible valeur de *max_feature* mène à des arbres moins corrélés les uns aux autres et donc à des forêts ayant une plus faible erreur quadratique moyenne.

5.4.5 Impact du nombre de prototypes

Pour étudier l'impact du nombre de prototypes sur les performances des modèles de forêt aléatoire, des expériences on été menées comme suit. Le jeu de données est divisé en une base d'entraînement de taille 500 et une base d'évaluation contenant les 1719 données restantes. L'heuristique CoreSelect est utilisée pour sélectionner 100 prototypes. Comme présenté annexe B, il s'agit d'une heuristique itérative gloutonne. Ainsi, tout groupe constitué de j premiers prototypes est un groupe de prototypes valide. Pour chaque j dans la liste $[20, 40, 60, 80, 100]$, trois modèles de forêts aléatoires contenant chacun 500 arbres sont entraînés. Chaque forêt correspond à une valeur différente du paramètre *max_feature* dans la liste $[0.1, 0.5, 1.0]$. Les trois forêts sont évaluées sur la base d'évaluation. Ces opérations sont répétées 30 fois pour 30 séparations aléatoires de la base de données entre une base d'entraînement et une base d'évaluation.

Les résultats de ces expériences sont présentés figure 5.2. Chaque boite à moustache correspond à un nombre de prototypes et une valeur du paramètre *max_feature* différents. Il est tout d'abord intéressant de noter que cette figure montre un fait qui avait déjà été observé précédemment. Des valeurs plus faibles du paramètre *max_feature* mènent à des valeurs plus faibles de la RMSE en réduisant la corrélation entre les arbres des forêts aléatoires. Ce fait s'observe d'autant plus pour des nombres importants de prototypes. Par ailleurs, il semble y avoir, ici, peu d'intérêt à augmenter le nombre de prototypes au delà de 40 ou 60. En effet, pour $max\_feature = 0.1$ la moyenne des RMSE passe seulement 1.765 à 1.760. Cependant, réaliser un test de student apparié pour comparer ces moyennes donne une p-value de 5.7×10^{-4}. On peut donc conclure, au risque 5%, que l'ajout des 40 prototypes supplémentaire permet une réduction, même légère, des RMSE sur ce jeu de données. Le même test réalisé entre 80 et 100 prototypes n'est, cependant, plus

Représentation des scans	**Modèle d'apprentissage**	s^{pre}	s^{pro}	$s^{pre \times pro}$	**précision**	**rappel**	F_1	RMSE
Métiers	**RF**	62.7 (0.8)	77.6 (0.7)	49.0 (0.8)	64.8 (0.8)	51.6 (1.2)	55.6 (0.9)	1.86 (0.04)
	MLP	56.0 (2.8)	76.1 (2.2)	41.4 (1.3)	61.4 (2.0)	43.9 (2.2)	47.5 (1.6)	2.09 (0.04)
ICP	**RF**	**63.7 (1.0)**	**78.4 (0.7)**	**50.4 (0.7)**	**66.4 (0.8)**	**53.3 (0.9)**	**57.3 (0.9)**	**1.79 (0.03)**
	MLP	61.5 (2.1)	75.3 (1.9)	45.6 (1.1)	62.9 (1.3)	49.8 (1.8)	53.2 (1.3)	1.94 (0.04)
ESF	**RF**	61.1 (0.9)	76.2 (0.6)	46.7 (0.7)	62.3 (0.8)	48.5 (0.9)	52.5 (0.8)	1.94 (0.03)
	MLP	59.3 (2.0)	76.5 (1.3)	45.0 (1.3)	62.9 (1.1)	47.3 (1.8)	51.7 (1.3)	1.96 (0.03)
M2DP	**RF**	49.5 (1.8)	70.9 (1.2)	33.5 (0.9)	47.4 (1.5)	28.7 (1.3)	33.3 (1.1)	2.49 (0.03)
	MLP	50.1 (3.1)	69.0 (2.3)	32.0 (1.3)	45.6 (1.8)	29.1 (2.0)	3.2 (1.6)	2.61 (0.05)

TABLE 5.1 : Scores d'évaluation moyens sur 30 répétitions des modèles de forêt aléatoire et de réseaux de neurones, pour plusieurs représentations structurées des grumes. Les écart-types des scores mesurés sur les 30 répétitions sont présentés entre parenthèses.

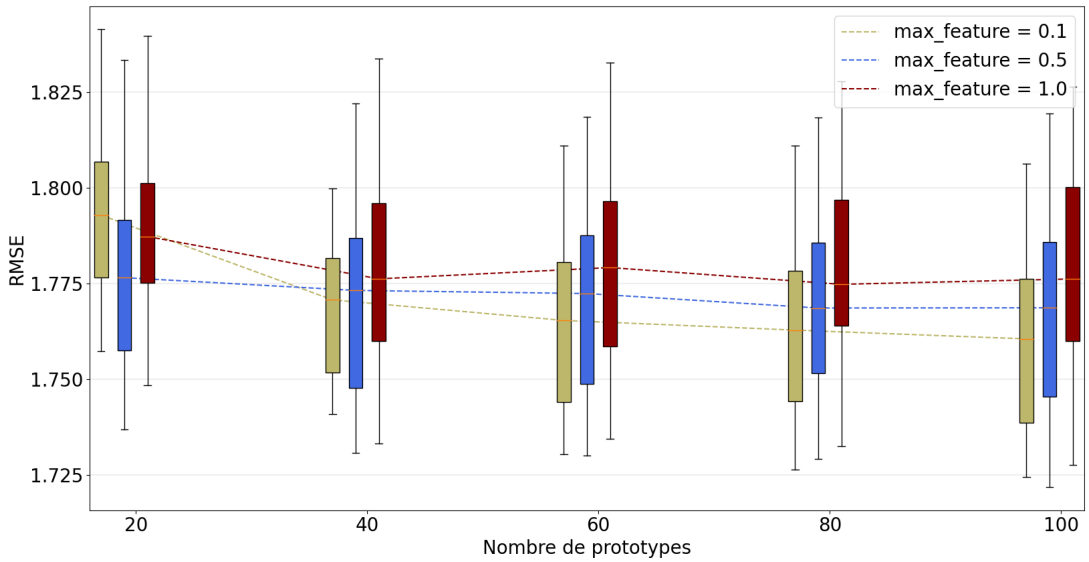

FIGURE 5.2 : Boites à moustaches des RMSE des modèles de forêts aléatoires pour 30 séparations entre base d'entraînement et base d'évaluation.

significatif.

Par ailleurs, un test de Student apparié comparant la moyenne des RMSE obtenus pour 100 prototypes avec max $\_feature = 0.1$ et max $\_feature = 0.5$ donne, ici, une p-value de 2.1×10^{-3}. Ainsi, la réduction de paramètre $max\_feature$ a un effet significatif sur les RMSE obtenues.

Puisque la RMSE moyenne la plus faible est obtenue pour $max\_feature = 0.1$ et 100 prototypes nous conserverons ces valeurs dans le reste de ce mémoire. Il est tout de même à noter qu'augmenter le nombre de prototypes augmente le coût en calcul de l'entraînement des modèles de forêt aléatoires et le coût en calcul des prédictions.

5.5 Couplage en ligne

5.5.1 Premiers résultats

L'algorithme d'apprentissage sélectionné dans la section 5.4 est intégré comme modèle de substitution pour les stratégies de couplages introduites chapitre 3 et évalué sur le jeu de données de résultats de simulation de sciage. Les quatre variantes basées sur les mesures d'incertitudes *Ambiguité*, *Variance d'échantillonnage*, *Consistance* et *Dissimilarité* sont à nouveau évaluées et comparées aux méthodes *Maximisation naïve* et 1 *in* ν. La mesure d'incertitude utilisée par la stratégie *Dissimilarité* est la dissimilarité ICP (symétrisée) directement, et non pas la distance dans le sous-espace de dissimilarité. Ainsi, en reprenant les notations du chapitre 3, la mesure d'incertitude de la stratégie *dissimilarité* est donnée par :

$$diss(S) = \min_{S_i in D} d_{ICP}(S, S_i) + d_{ICP}(S_i, S)\,. \tag{5.11}$$

Dans cette section, les flux d'arrivée utilisés restent stationnaires. Les erreurs préquentielles de couple associées à chaque stratégie sont calculées comme définit dans le chapitre 3. Cependant, en plus de l'erreur préquentielle utilisant l'erreur quadratique moyenne, nous calculons également les scores préquentiels des couples associés aux scores F_1 et $s^{pre\times pro}$. En plus des scores et erreurs préquentielles des couples, les scores et erreurs préquentielles du modèle de substitution seul sont également calculés. Ils correspondent à la moyenne au cours du temps des erreurs (ou scores) de toutes les prédictions réalisées par le modèle de substitution dans le passé du flux, qu'une simulation ait été lancée ou non. Ces erreurs prequentielles seront utilisées pour évaluer l'impact des différentes méthodes d'échantillonnage sur les performances du modèle de substitution seul lorsque celui-ci sera réentraîné sur les bases complétées.

Dans un premier temps, cependant, les modèles de substitution ne sont entraînés qu'une seule fois en début de flux. Les erreurs et scores préquentielles des couples modèle de simulation-modèle de substitution pour les six stratégies de couplage évaluées durant ces expériences sont présentés table 5.2. Ces résultats sont moyennés sur 100 répétitions des expériences.

Concernant l'erreur quadratique préquentielle du couple simulateur-modèle de substitution, les meilleurs résultats sont, encore une fois, obtenus pour la stratégie basée sur la mesure *ambiguité*. L'erreur quadratique du couple associée à cette stratégie est, en particulier, plus faible que pour la stratégie de *maximisation naïve* malgré le moindre nombre de simulations réalisées. Un test de comparaison de Student entre la moyenne des erreurs préquencielles MSE obtenues au cours des 100 répétitions effectuées avec la stratégie utilisant la mesure *ambiguïté* et celle obtenue avec la stratégie de *maximisation naïve* donne une p-value de 1.5×10^{-39}. Ainsi, cette différence est statistiquement significative.

Ce n'est, cependant, pas le cas pour les scores $s^{pre\times pro}$ et F_1. Dans ce cas, les quatre stratégies échantillonnant activement le flux ont des scores plus faible que la stratégie de *maximisation naïve*, voir même légèrement plus faible que les scores de la stratégie 1 in ν pour la *consistance*. En particulier la réalisation de test de Student pour comparer les moyennes de ces scores mesurés pour chacune des stratégies utilisant une des quatre mesures d'incertitude avec la moyenne des scores pour la stratégie de *maximisation naïve* donne (presque) toujours des p-values inférieures à 10^{-7}. Ainsi, les différences observées sont significatives. La seule exception est le test comparant la moyenne des scores F_1 obtenus avec la stratégie utilisant la mesure *ambiguïté* avec ceux obtenus avec la stratégie de *maximisation naïve*. Dans ce cas, la p-value est égale à 0.6. Ces scores plus faibles peuvent s'expliquer par le fait que la mesure *ambiguïté*, en particulier, est moins corrélée avec les scores $s^{pre\times pro}$ et F_1 qu'avec l'erreur quadratique. Si l'objectif était de minimiser ces scores, d'autres mesures devraient être utilisées.

En outre, toutes les stratégies ne mènent pas à une réduction de l'erreur quadratique préquentielle. Les stratégies utilisant les mesures *ambiguïté* et *variance d'échantillonnage* produisent bien des RMSE significativement plus faible en moyenne, comme montré encore une fois par des tests de Students. Un test de Student comparant les moyennes des erreurs quadratiques préquentielles obtenues pour la stratégie *consistence* avec la stratégie de *maximisation naïve* donne, en effet, une p-values de 1.0×10^{-4} indiquant que la stratégie *consistance* mène à des erreurs prequentielles plus hautes en moyenne. Ces erreurs préquentielles restent, cependant, significativement plus faibles en moyenne que celles de la stratégie 1 in ν. De même, la p-value du test comparant la

moyenne des erreurs préquentielles obtenues pour les stratégies *dissimilarité* et *maximisation naïve* est de 8% ce qui rend difficile de trancher. Cependant, la p-value de test comparant les stratégies *dissimilarité* et 1 in ν donne une p-value de 2.5×10^{-12}. Toutes les stratégies utilisant une des mesures d'incertitudes ont, plus particulièrement, des erreurs préquentielles plus faibles en moyenne que la référence 1 in ν.

	Erreur préquentielle du couple		
Stratégie	$\mathbf{F_1}$	$\mathbf{s^{pre \times pro}}$	**quadratique**
1 in ν	66.6 (0.5)	60.8 (0.5)	2.46 (0.07)
Maximisation naïve	**68.6 (0.8)**	**63.1 (0.8)**	2.33 (0.09)
Ambiguïté	68.2 (0.9)	62.7 (0.8)	**2.14 (0.06)**
Variance d'échantillonnage	67.5 (0.8)	62.0 (0.8)	2.28 (0.08)
Consistance	65.2 (0.8)	59.9 (0.9)	2.38 (0.08)
Dissmilarité	66.2 (0.8)	60.5 (0.8)	2.31 (0.06)

TABLE 5.2 : erreurs et scores préquentielles du couple modèle de simulation-modèle de substitution pour les 6 stratégies de couplage sur les jeux de données de résultats de simulations de sciage. Les scores sont moyennés sur 100 répétitions des expériences. Leurs écart-types mesurés sur ces répétitions sont également donnés. Le modèle d'apprentissage n'est entraîné qu'une seule fois en début de flux.

La table 5.3 présente les résultats d'expériences similaires à ceux de la table 5.2. Cependant, cette fois les modèles de substitutions sont re-entraînés à chaque fois que 50 nouveaux résultats de simulations sont obtenus. Les résultats sont comparables à ceux présentés dans la table 5.2. La plus forte réduction de l'erreur préquentielle quadratique est observée pour la stratégie utilisant la mesure *ambiguïté*. Cette fois, cependant, on observe une erreur quadratique préquentielle du modèle de substitution seul plus faible pour les stratégies de couplage utilisant les mesures d'incertitude que pour les stratégies de référence. La réduction la plus forte est, encore une fois, observée pour la mesure *ambiguïté*. En particulier, un test de Student comparant la moyenne des MSE obtenues par la stratégie utilisant la mesure *ambiguïté* sur les 100 répétitions de l'expérience

avec la moyenne des MSE obtenues avec la stratégie de maximisation naïve donne une pvalue de 1.3×10^{-10}. Les points de données échantillonnés par les mesures d'incertitudes permettent donc d'entraîner des modèles d'apprentissage plus performant. Ceci est, en effet, une hypothèse de l'apprentissage actif qui n'est cependant pas toujours vérifiée en pratique pour tous les problèmes de prédiction, méthodes d'échantillonnage et taux d'échantillonnage.

Stratégie	**Couple Simulateur-Modèle de substitution**	**Modèle de substitution seul**
1 in ν	2.39 (0.06)	2.99 (0.06)
Maximisation naïve	2.24 (0.07)	2.97 (0.05)
Ambiguïté	**2.07 (0.06)**	**2.92 (0.05)**
Variance d'échantillonnage	2.19 (0.06)	2.95 (0.06)
Consistance	2.35 (0.06)	2.98 (0.07)
Dissmilarité	2.24 (0.06)	2.94 (0.05)

TABLE 5.3 : Erreur préquentielle quadratique des 6 stratégies de couplage sur le jeu de données de résultats de simulations de sciage. Les scores sont moyennés sur 100 répétitions des expériences. Leurs écart-types mesurés sur ces répétitions sont également donnés. Le modèle d'apprentissage est re-entraîné chaque fois que 50 nouvelles données d'apprentissage sont disponibles.

5.5.2 Impact de $\frac{\lambda}{\mu}$

Comme montré dans le chapitre 3, le taux d'échantillonnage du flux et la propension de méthodes de couplage à avoir une erreur préquentielle de couple plus faible qu'une simple méthode de *maximisation naïve* dépendent fortement du paramètre $\frac{\lambda}{\mu}$, c'est à dire du rapport entre le temps moyen de simulation et le temps moyen d'arrivée entre deux points de données. L'objectif de cette section est de valider expérimentalement l'influence de ce paramètre sur l'erreur préquentielle quadratique.

En particulier plusieurs expériences numériques ont été menées comme dans la section précédente mais en augmentant progressivement la moyenne des temps générés pour émuler les temps de simulations de 20 à 100. Ces résultats sont présentés table 5.4. Ils confirment ce qui avait été

inféré dans le chapitre 3. Lorsque le temps moyen de simulation est faible, pour $\frac{\lambda}{\mu} = 20$, les taux d'échantillonnages de la méthode basée sur l'apprentissage actif et de la stratégie de *maximisation naïve* sont tous les deux grands, mais l'écart entre leurs taux d'échantillonnage est également grand. Ainsi, la stratégie de *maximisation naïve* utilise beaucoup plus le modèle de simulation et, en conséquence, présente une erreur préquentielle du couple similaire à celle obtenue avec la stratégie basée sur l'apprentissage actif, et ce malgré un échantillonnage du flux ne considérant pas l'incertitude des prédictions. En effet, un test de student comparant les erreurs préquentielles obtenues sur les 30 répétitions pour chacune des deux stratégies donne une p-value de 0.61. La différence n'est donc pas, ici, significative. Le même test réalisé pour les autres valeurs de $\frac{\lambda}{\mu}$ donne, cependant, toujours des p-values inférieurs à 10^{-29}, soit virtuellement zéro. Lorsque le temps de simulation moyen augmente, en effet, les taux d'échantillonnages des deux stratégies diminuent et l'écart entre les deux se réduit, avec le taux d'échantillonnage de la stratégie basée sur l'apprentissage actif restant plus faible que pour la stratégie naïve. Ainsi, pour tous les temps de simulation moyen évalués supérieurs à 40, la stratégie proposée a une erreur préquentielle quadratique du couple plus faible que celle de la stratégie par *maximisation naïve*.

	Ambiguïté		**Maximisation naïve**	
$\frac{\lambda}{\mu}$	**Taux d'échantillonnage**	**erreur quadratique prequentielle**	**Taux d'échantillonnage**	**erreur quadratique prequentielle**
20	0.33 (0.01)	1.66 (0.06)	0.46 (0.02)	**1.65 (0.07)**
40	0.20 (0.01)	**2.13 (0.07)**	0.25 (0.01)	2.33 (0.08)
60	0.15 (0.01)	**2.37 (0.06)**	0.17 (0.01)	2.56 (0.09)
80	0.12 (0.01)	**2.52 (0.07)**	0.13 (0.01)	2.70 (0.09)
100	0.10 (0.01)	**2.60 (0.08)**	0.10 (0.01)	2.78 (0.10)

TABLE 5.4 : Erreur quadratique préquentielle des couples modèle de simulation-modèle de substitution. La stratégie de couplage est basée sur la mesure *ambiguïté*. Des valeurs croissantes du temps moyen de simulation sont émulées. Les scores sont moyennés sur 30 répétitions des expériences. Leurs écart-types mesurés sur ces répétitions sont également donnés. Le modèle d'apprentissage n'est entraîné qu'une seule fois en début de flux.

5.5.3 Détection de dérives

Cette section étudie et compare les performances des méthodes de détection de dérives introduites au chapitre 3 dans le cas du simulateur de sciage étudié. En plus du cas de la dérive réelle étudié précédemment, le cas de la dérive virtuelle est également considéré.

Pour générer les flux avec dérive réelle, un jeu de données additionnel constitué de résultats de simulations pour une configuration de scierie différente de celle utilisée pour le jeu de données décrit dans la section 5.3 a pu être obtenu pour 368 grumes. Considérant la faible taille des jeux de données considérés, les flux sont générés en tirant aléatoirement, avec remise, 5000 grumes dans le jeux de données de la section 5.3 et 5000 grumes dans le jeu de données additionnel. L'utilisation de tirage avec remise, cependant, implique que le modèle d'apprentissage ne peut pas être réentrainés en cours de flux sous peine de biaiser les résultats. Les 500 exemples d'entraînement utilisés pour l'initialisation des modèles de substitution ne sont pas réutilisés dans les flux.

Les résultats expérimentaux pour ce type de dérive réelle sont présentés dans la table 5.5. Cette table présente les taux de détection, nombre moyen de fausses alertes et temps médian de détection des dérives pour 100 répétitions des expériences avec les différentes méthodes de détections introduites au chapitre 3. Les colonnes nommées "AL" correspondent aux expériences ou le flux est échantillonné avec la mesure *ambiguité*. Les colonnes nommées "RD" correspondent aux expériences ou cette mesure est remplacée par une valeur aléatoire uniforme.

Une observation importante est que les différentes méthodes de détections de dérive sur ce jeu de données sont toutes affectées par le biais d'échantillonnage. En particulier, les tests de Brunner-Munzel comparant les temps de détections obtenus en échantillonnant les flux au hasard ou avec la mesure ambiguïté ont des p-values inférieures à 5% pour toutes les méthodes de détections testées. La plus grande p-value, 0.047, est obtenue pour la méthode de Kolmogorov-Smirnov avec $\rho = 0.001$. Le biais d'échantillonnage impacte également les taux de détection. En particulier, des tests de fisher exacts comparant les taux de détection obtenus en échantillonnant au hasard ou avec la mesure ambiguïté ont des p-values inférieures à 5% pour toutes les méthodes de détections testées sauf pour celle basée sur les cartes de contrôles. La méthode de Page-Hinkley, en particulier, présente des taux de détection très bas en présence de biais d'échantillonnage. Bien qu'elle présente les temps de détection médians les plus faibles pour le cas de l'échantillonnage aléatoire du flux, elle ne saurait donc être recommandée en conjonction avec un échantillonnage actif.

En général, les méthodes de Kolmogorov-Smirnov avec $\rho = 0.005$ et Anderson-Darling avec $\rho = 0.005$ semble présenter un bon équilibre entre taux de détection et temps de détection. Anderson-Darling semble présenter, sur ces expériences, un temps de détection médian plus faible. La différence n'est cependant pas significative pour le test de Brunner-Munzel avec une p-value de 0.22. Ces méthodes ont, par ailleurs, un taux de fausses alertes très faible.

Le cas de la dérive virtuelle est également étudié. Pour ce faire, la base de donnée est divisée en deux parts égales suivant la longueur des grumes. Les flux générés sont composés d'abord de 5000 grumes tirées au hasard avec remise parmi les grumes courtes (dont sont omises les grumes ayant servi à l'entraînement du modèle de substitution) puis de 5000 grumes tirées au hasard parmi les grumes longues.

Les résultats sont présentés dans la table 5.6. Les résultats sont très différents du cas de la dérive réelle. En effet, les temps de détection sont, cette fois-ci, significativement plus faible pour les

		Taux de détection		Nbr fausses alertes		Temps de détection	
		AL	**RD**	**AL**	**RD**	**AL**	**RD**
Kolmogorov-Smirnov	$\rho =$ 0.005	0.92	0.99	0.32 (0.56)	0.26 (0.48)	758 (907)	451 (210)
	$\rho =$ 0.001	0.75	0.97	0.11 (0.31)	0.06 (0.24)	719 (1056)	632 (648)
Anderson-Darling	$\rho =$ 0.005	0.84	1.0	0.27 (0.55)	0.15 (0.38)	586 (974)	474 (212)
	$\rho =$ 0.001	0.73	0.98	0.09 (0.29)	0.02 (0.14)	699 (1088)	525 (426)
Page-Hinkley	$h = 10\sigma$	0.34	0.73	1.69 (1.40)	2.19 (1.80)	848 (1374)	291 (519)
	$h = 15\sigma$	0.28	0.78	0.77 (0.94)	0.85 (0.97)	754 (2037)	295 (418)
Cartes de controle		0.96	1.0	8.40 (2.71)	7.41 (1.98)	482 (629)	325 (563)

TABLE 5.5 : Taux de détection, nombre moyen de fausses alertes et temps médian de détection pour différentes méthodes de détection de dérives dans le cas de la dérive réelle. Les valeurs entre parenthèses correspondent aux écart-types pour les nombres de fausses alertes et aux écarts inters-quartiles pour les temps de détection.

flux échantillonnés à l'aide de la mesure *ambiguïté* que pour les flux échantillonnés au hasard. Cela s'explique par le fait que, même après l'instant de dérive, l'échantillonnage actif continue à sélectionner des points de données pour lesquels l'erreur de prédiction est, en moyenne, grande, ce qui accélère la détection. Pour illustration, la figure 5.3 présente les histogrammes des erreurs quadratiques des points de données échantillonnés pour simulation, mesurées après l'instant de dérive. L'histogramme rouge correspond aux erreurs mesurées pour un flux échantillonné activement par la mesure *ambiguïté* et l'histogramme bleu à un flux échantillonné au hasard. Ainsi, dans cet exemple, la mesure *ambiguïté* est toujours capable de détecter des points de données pour lesquels l'erreur de prédiction est importante, même après dérive. Cela contrebalance l'effet du biais d'échantillonnage et favorise la détection de ces dérives virtuelles. Il n'est, cependant, pas garantit que ce comportement se reproduise pour d'autres types de dérives et jeux de données.

Dans ce cas précis, les temps de détection les plus faibles sont observés pour la méthode de Page-Hinkley. En effet, le test de Brunner-Munzel comparant, par exemple, la distribution des temps de détection entre Page-Hinkley avec $h = 15\sigma$ et Anderson-Daring avec $\rho = 0.005$ donne une p-value de virtuellement 0. Ces méthodes présentent, cependant, un nombre moyen de fausses alertes plus grand que les méthodes Kolmogorov-Smirnov et Anderson-Darling. Un test de Student comparant ces moyennes donne, en particulier, une p-value de 3.3×10^{-5}. De même, Page-Hinkley présente un taux de détection plus faible que les méthodes Kolmogorov-Smirnov et Anderson-Darling qui détectent, ici, la totalité des dérives.

Il pourra être remarqué que la méthode basée sur les cartes de contrôle présente, quel que soit le type de dérive, un nombre de fausses alertes particulièrement important par rapport aux autres méthodes, que le flux soit échantillonné activement ou non. Cependant, les conditions de rejet des cartes de contrôles sont définies sous l'hypothèse que le flux contrôlé suit une loi normale. Cette condition est loin d'être vérifiée ici comme illustré par les histogrammes présentés figure 5.3.

		Taux de détection		Nbr fausses alertes		Temps de détection	
		AL	**RD**	**AL**	**RD**	**AL**	**RD**
Kolmogorov-Smirnov	$\rho = 0.005$	1.0	1.0	0.42 (0.62)	0.29 (0.49)	177 (55)	348 (155)
	$\rho = 0.001$	1.0	1.0	0.12 (0.34)	0.09 (0.29)	189 (63)	416 (132)
Anderson-Darling	$\rho = 0.005$	1.0	0.99	0.39 (0.60)	0.28 (0.53)	169 (57)	336 (132)
	$\rho = 0.001$	1.0	1.0	0.13 (0.36)	0.09 (0.29)	185 (65)	376 (158)
Page-Hinkley	$h = 10\sigma$	0.97	0.97	1.67 (1.45)	1.83 (1.51)	65 (56)	106 (119)
	$h = 15\sigma$	0.92	0.94	1.01 (1.32)	0.97 (1.17)	72 (50)	116 (113)
Cartes de controle		0.96	1.0	5.63 (1.83)	8.04 (1.77)	114 (322)	209 (514)

TABLE 5.6 : Taux de détection, nombre moyen de fausses alertes et temps médian de détection pour différentes méthodes de détection de dérives dans le cas de la dérive virtuelle. Les valeurs entre parenthèses correspondent aux écart-types pour les nombres de fausses alertes et aux écarts inters-quartiles pour les temps de détection.

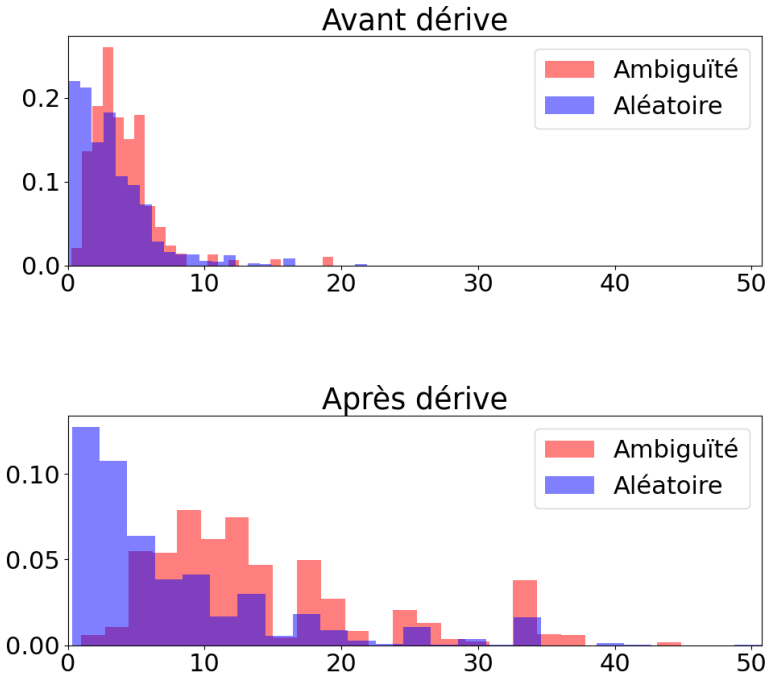

FIGURE 5.3 : Histogrammes des erreurs quadratiques commises par le modèle de substitution avant et après l'instant de dérive pour deux méthodes d'échantillonnage du flux. La première utilise la mesure *ambiguïté*, la deuxième la "mesure" aléatoire.

5.6 Conclusion

Ce chapitre présente une application de la stratégie de couplage introduite au chapitre 3 pour le développement d'une ombre numérique de scierie constituée d'un simulateur de sciage et d'un modèle de simulation de ce simulateur. L'objectif de cette ombre numérique est la prédiction du panier de produits de grumes de bois à partir de scans 3D de leur forme.

> Un nouveau modèle de substitution utilisant la notion de plongement dans un sous-espace de dissimilarité pour la construction d'un vecteur de caractéristiques servant d'entrée à un modèle de forêt aléatoire est d'abord proposé et motivé. Ce modèle de substitution est ensuite utilisé dans le cadre de la stratégie de couplage. Les quatre mesures d'incertitudes des prédictions introduites précédemment ont été évaluées sur ce jeu de données de résultats de simulation et la plus forte réduction de l'erreur quadratique préquentielle est observée pour la stratégie utilisant la mesure *ambiguité* qui est, en particulier, plus faible que l'erreur obtenue pour la méthode *maximisation naïve*.

L'impact du rapport entre le temps moyen de simulation et le temps moyen d'inter-arrivées entre deux points de données du flux est également évalué, et il est montré que pour des temps de simulation moyens plus faibles, la stratégie *maximisation naïve* l'emporte, du fait de la différence importante entre les taux d'échantillonnage des stratégies comparées. Cependant, la stratégie proposée présente un avantage dès lors que les temps moyens de simulation sont suffisamment grands.

Plusieurs méthodes de détections de dérives sont également évaluées dans le cadre de dérives réelles et de dérives virtuelles. Il est montré que dans le cas de la dérive réelle, les méthodes de détection appliquées à ce jeu de données sont très sensibles au biais d'échantillonnage qui a un impact négatif sur les taux et temps de détection dans le cas de la dérive réelle. Cependant, dans le cas de la dérive virtuelle, le fait que la mesure d'ambiguïté reste capable d'identifier les points de données pour lesquels l'erreur de prédiction est importante même après la dérive bénéficie à ces méthodes de détection.

Chapitre 6

Limites et perspectives

6.1 Introduction

Ce mémoire de thèse présente une stratégie de couplage entre un modèle de simulation et son modèle de substitution pour le développement d'O&JN. De nombreuses expériences numériques ont été réalisées. Ces expériences ont, en particulier, permis de démontrer les bénéfices de la stratégie proposée en terme de réduction de l'erreur préquentielle des prédictions. Cependant, elles ont également permis de démontrer une première limite de la méthode proposée : l'effet négatif du biais d'échantillonnage sur la détection de dérives conceptuelles réelles. Bien que le cas de la dérives réelles ne soit pas le plus vraisemblable dans le contexte d'utilisation présenté ici, il s'agit d'un point de vigilance à considérer.

Les limites de la stratégie proposée et développée dans ce mémoire sont détaillées dans la suite de ce chapitre. Des pistes de travaux futur concernant l'amélioration de la stratégie proposée, son analyse théorique et son utilisation pratique sont également développées.

6.2 Limites de la stratégie proposée

La stratégie de couplage proposée dans ce mémoire de thèse est inspirée de l'apprentissage actif. Le flux de données n'est pas échantillonné aléatoirement et indépendamment des caractéristiques des données transmises. Cela induit un biais d'échantillonnage, c'est à dire une divergence entre la distribution statistique des données transmises par le flux et la distribution statistique des données échantillonnées et labélisées par le modèle de simulation qui sont utilisables pour l'entraînement et l'amélioration du modèle de substitution. L'effet de ce biais d'échantillonnage sur la détection de dérives a été étudié dans ce mémoire. Cependant, il induit, en pratique d'autres limitations qui doivent être mentionnées. L'une des plus importantes est que ce biais peut mener, dans certains cas, à des modèles d'apprentissage moins performants que s'ils avaient été entraînés sur un jeu de données échantillonnées au hasard, par exemple en négligeant l'échantillonnage de certaines zones de l'espace des caractéristiques. Par ailleurs, ce biais complique, en usage réel, l'évaluation des performances du modèle de substitution. En particulier, l'erreur préquentielle mesurée sur

sur les données effectivement labélisées par le simulateur n'est pas représentative de l'erreur préquentielle du couple, ou de l'erreur préquentielle des données non échantillonnées. Cela signifie que le choix du type d'algorithme d'apprentissage utilisé pour créer le modèle de substitution ainsi que l'optimisation de ces hyperparamètres ne peut être fait sur la base échantillonnée activement.

D'autres limites de la stratégie de couplage présentée dans ce mémoire sont induites par les hypothèses utilisées. En particulier, une hypothèse considérée tout au long de cette thèse est l'indépendance entre l'erreur de prédiction et le temps de simulation associé à un même point de données. Les vrais temps de simulation étant absents des jeux de données considérés, ils sont remplacés par des variables aléatoires indépendantes des points de données dans les expériences présentées ici. Il est cependant envisageable que des points de données pour lesquels l'erreur de prédiction soit plus grande nécessitent également des temps de simulation plus long. Dans ce cas, le taux d'échantillonnage du flux serait réduits par rapport aux résultats présentés dans ce mémoire. Si l'objectif reste de diminuer l'erreur préquentielle quadratique du couple, c'est à dire la *moyenne* temporelle des erreurs quadratiques, la question de l'impact relatif de la réduction du taux d'échantillonnage par rapport à la réduction de l'erreur des prédictions des données pour lesquelles aucune prédiction n'est lancée devient plus complexe. Cette question déjà posée par la comparaison entre la méthode de *maximisation naïve* et les méthodes basées sur l'apprentissage actif est cependant centrale dans la stratégie proposée. Dans le cas évoqué ici, la décision de lancer une simulation ou non pour un point de données devrait, en particulier, tenir compte du temps de simulation estimé.

Une deuxième hypothèse forte réalisée est que le modèle de simulation est suffisamment précis pour assimiler la prédiction de ce modèle au vrai label des données et simplifier ainsi l'expression de l'erreur préquentielle du couple. De plus, le modèle de substitution est entraîné comme une approximation du modèle de simulation. Si les prédictions du modèle de simulation servant à l'entraînement du modèle de substitution sont bruitées, les prédictions de celui-ci seront d'autant plus éloignées du vrai label. Si le cas d'utilisation le permet, il serait plus intéressant d'entraîner le modèle de substitution sur les vrai labels, ce qui entraîne l'introduction d'un autre oracle, différent du modèle de simulation, par exemple un expert humain.

6.3 Travaux futur

6.3.1 Amélioration de la stratégie de couplage

De nombreuses pistes d'amélioration peuvent être étudiées pour approfondir l'étude de la stratégie proposée et la rendre mature pour une utilisation industrielle. L'une des plus importantes est la gestion de la base d'exemples collectée pour l'entraînement et le ré-entraînement du modèle de substitution. Tel que présenté dans ce mémoire, tous les résultats de simulations obtenus au cours du temps sont ajoutés à cette base d'entraînement et n'en sont jamais retirés. De fait, la taille de la base d'entraînement ne peut que croître, ce qui à terme aura un impact non négligeable sur le temps d'entraînement du modèle d'apprentissage sans pour autant contribuer à réduire d'avantage son erreur préquentielle. Cela est d'autant plus problématique en présence de dérive conceptuelle, lorsque certaines données deviennent obsolètes.

Plusieurs pistes de travail peuvent être considérées pour palier à ce problème. La première est l'utilisation de modèles d'apprentissage incrémentaux. Ces modèles sont mis à jour à chaque

nouvelle arrivée d'un point de données labélisé, sans qu'il soit nécessaire de les ré-entraîné complètement. Considérant les bonnes performances des forêts aléatoires dans le cas des scieries, l'utilisation de modèles incrémentaux utilisant également des structures d'arbres, comme les arbres de Hoeffding (Gama, Rocha et Medas, 2003), ou les forêts de Mondrian (Lakshminarayanan, Roy et Teh, 2014), parait une piste prometteuse. De même, le modèle SAMkNN (Losing, Hammer et Wersing, 2016) parait intéressant dans le contexte plus général de l'apprentissage par dissimilarité. Il s'agit, en effet, d'une variante incrémentale de l'algorithme des plus proches voisins capable de réagir à différents types de dérives conceptuelles grâce à une distinction entre mémoires à court et long termes.

Une deuxième solution est l'utilisation explicite de mécanismes d'oublis permettant d'éliminer certains exemples de la base d'entraînement pour en limiter la taille. L'une des méthodes les plus simples est l'élimination des exemples les plus anciens, mais de nombreuses alternatives ont été proposées, comme la réduction de la base de données par la sélection de prototypes. Le modèle SAMkNN, par exemple, utilise un mélange de ces méthodes. La "mémoire à court terme" est gérée suivant une fenêtre glissante, tandis que la "mémoire à long terme" est compressée en utilisant un algorithme de clustering sur les données et en ne gardant que les centres de classe comme prototypes. Une méthode alternative serait l'utilisation de méthodes d'apprentissage actif pour la compression de ces bases d'entraînement.

D'autres travaux futurs pourront s'intéresser à deux éléments centraux de la stratégie de couplage que sont la mesure d'incertitude et la fonction g gouvernant l'évolution du seuil auquel est comparée la mesure d'incertitude en fonction du taux d'utilisation des serveurs de simulation. En particulier, il pourrait être intéressant de combiner plusieurs des mesures d'incertitudes étudiées dans ce mémoire pour améliorer les performances de la stratégie de couplage. De même, la fonction g principalement utilisée dans ce mémoire est une fonction linéaire du taux d'utilisation des serveurs, mais d'autres fonctions, comme des fonctions puissances, pourraient par exemple être utilisées. L'utilisation d'une fonction g concave permettrait, en effet, d'augmenter les seuils par rapport à la fonction linéaire utilisée ici, et mènerait à un taux d'échantillonnage du flux plus faible, tandis qu'une fonction g convexe aurait l'effet inverse.

6.3.2 Analyse des propriétés théoriques

L'analyse des propriétés théoriques du système dynamique induit par la stratégie de couplage qui est détaillée dans le chapitre 3 reste introductive. Deux points, en particulier, méritent d'être approfondis.

Le premier est l'amélioration de la borne inférieure à la réduction de l'erreur préquentielle de couple pouvant être obtenu par la stratégie de couplage. Cette borne inférieure fait, en particulier, intervenir la corrélation entre la variable binaire représentant la décision de lancer ou non une simulation pour un point de données et l'erreur commise par le modèle de substitution. Cependant, le lien entre la mesure d'incertitude et cette variable de décision reste peu explicite. Une étude plus fine des liens entre la mesure d'incertitude, l'erreur commise par le modèle de substitution et la décision de lancer ou non une simulation pourra amener à une borne plus réaliste et à une meilleure compréhension du lien entre la mesure d'incertitude et la réduction de l'erreur préquentielle.

Par ailleurs, une hypothèse fondamentale de toute l'analyse théorique réalisée dans ce mémoire

est l'ergodicité des systèmes dynamiques étudiés. Dans les travaux présentés ici, l'ergodicité n'est assurée que dans le cas Markovien, c'est à dire lorsque les temps d'inter-arrivées et de simulation suivent des lois exponentielles. Ce cas reste très limité en pratique. En particulier, les temps de simulation observés pour quelques simulations de sciage réalisées ne suivent pas une loi exponentielle. Pourtant, les taux de simulation observés sont cohérents avec les résultats obtenus sous hypothèse d'ergodicité. Ainsi, des hypothèses plus large sur le comportement des temps de simulation garantissant l'ergodicité du système dynamique étudié est souhaitable.

6.3.3 Utilisation pratique

Au moins deux utilisations pratiques d'une ombre numérique telle que détaillé dans ce mémoire de thèse peuvent être distinguées. La première est l'utilisation des données générées par l'ombre numérique par des outils d'aide à la décision, après agrégation par lot et à des intervalles de temps grand devant les temps d'inter-arrivées et de simulation. Dans ce cas, le délai entre le passage d'un point de données dans le flux et le temps d'obtention de la prédiction si le simulateur est utilisé n'est pas une contrainte. Dans le contexte des scieries, ce cas se retrouve si les prédictions des paniers de produits sont agrégées par classes ou lot de grumes et utilisées pour optimiser un plan de production ou de distribution. C'est, par exemple, ce qui est fait dans Morin et al., 2020 qui utilise des modèles de substitution pour estimer le rendement de lots de grumes dans plusieurs scieries et optimiser leur distribution. Cela nécessite, cependant, une traçabilité fine des grumes qui peut être problématique en scierie.

Le deuxième cas d'utilisation à distinguer se retrouve lorsqu'une décision doit être prise pour chaque point de données suite à la prédiction réalisée, par exemple un paramétrage du processus de transformation associé au point de données. Dans le cas des scieries, il pourrait être souhaitable d'utiliser ces prédictions pour classer les grumes suivant leurs paniers de produits. Dans ce cas il est nécessaire de mettre en place une zone tampon pour stocker les produits pour lesquels une simulation est lancée, le temps que le résultat soit obtenu. La taille de cette zone tampon pourra être une contrainte plus forte que la capacité calculatoire allouée comme limite au nombre de simulations pouvant être lancées en parallèle à tout instant.

Ainsi, l'approfondissement de ces deux cas d'études pourra également faire l'objet de travaux supplémentaires.

6.4 Conclusion

Ce chapitre détaille les limites associées au modèle d'ombre numérique développé dans ce mémoire de thèse et liste des pistes de travaux futurs. Beaucoup des limites énoncées sont, en particulier, associées à l'utilisation de méthodes inspirées de l'apprentissage actif. D'autres limites sont associées aux hypothèses de travail utilisées dans ce mémoire.

Par ailleurs, trois axes de travaux de recherches supplémentaires ont été explicités dans ce chapitre. Le premier concerne l'amélioration de la stratégie de couplage elle-même, par exemple en combinant plusieurs mesures d'incertitudes ou en étudiant la gestion de la base d'entraînement du modèle d'apprentissage automatique. Le deuxième axe se concentre sur l'analyse théorique des propriétés de la stratégie de couplage. Enfin, le troisième axe concerne l'utilisation des prédictions réalisées par l'ombre numérique pour l'aide à la décision.

Chapitre 7

Conclusion générale

Les travaux réalisés durant cette thèse étudient les ombres et jumeaux numériques. De nombreux concepts et définitions sont associés à ces deux termes. Dans ce mémoire, les ombres numériques sont, en particulier, définies comme un ensemble de modèles numériques pour l'aide à la décision traitant un flux de données transmit par un homologue physique. La contribution principale de ce mémoire est la proposition d'une stratégie de couplage entre deux modèles numériques traitant la même tâche de prédiction mais ayant des niveaux de fidélité, des temps d'exécution et des prérequis en données différents.

Le premier chapitre de ce mémoire présente une analyse de la littérature traitant des ombres et jumeaux numériques. Ce travail est rendu d'autant plus nécessaire par la pluralité des définitions et usages associées aux concepts d'ombres et jumeaux numériques, ainsi que par leurs nombreuses catégorisations. En particulier, deux écoles de pensées sous-jacentes à ces définitions sont présentées. Plusieurs autres caractérisations sont également détaillées. L'objectif premier de cette étude de la littérature est de préciser quel type d'O&JN est considéré dans ce mémoire de thèse. De fait, les travaux présentés ici s'appuient sur la vision des O&JN comme une évolution des modèles de simulation, et plus généralement des modèles numériques. L'objectif est d'intégrer ces modèles numériques au cœur des systèmes, processus, ou produits, pour un usage opérationnel. Contrairement à la vision considérant les O&JN comme une évolution des modèles de conception assistée par ordinateur, le niveau de fidélité des modèles intégrés à l'O&JN n'est pas au cœur de l'approche considérée ici. Ainsi, les O&JN considérés dans cette dissertation sont des ensembles de modèles numériques. Ces modèles traitent des données collectées au cours du temps depuis leur homologue physique pour associer, notamment, une prédiction à chaque point de données. Parmi les nombreux verrous identifiés dans la littérature par ces O&JN, cette thèse se concentre sur la coordination efficiente de plusieurs modèles numériques disposant de niveaux de fidélités et de coûts en calcul différents.

Cette première analyse de la littérature est suivie d'une présentation des outils d'apprentissage automatique utilisés dans la suite de ce mémoire, ainsi que de leur lien avec les ombres et jumeaux numériques. En particulier, sont détaillés l'usage de modèles d'apprentissage comme modèle de substitution, l'apprentissage actif et la détection de dérives conceptuelles.

Le deuxième chapitre de cette dissertation introduit la contribution centrale des travaux effectués durant cette thèse : la proposition d'une stratégie de couplage entre un modèle de simulation et un modèle de substitution entraîné par apprentissage automatique. Le modèle de simulation est supposé avoir un haut niveau de fidélité mais être coûteux en temps de calcul. Le modèle d'apprentissage automatique est une approximation du modèle de simulation. Au contraire de ce dernier, il est rapide à exécuter sur de nouveaux points de données. Cependant, les erreurs de prédiction commises sont plus importantes que que pour le modèle de simulation.

L'objectif premier de la stratégie de couplage proposée est la réduction de l'erreur préquentielle commise par le couple de modèles. Pour cela, il doit être décidé, pour chaque point de données transmit depuis l'homologue physique, si la prédiction du modèle d'apprentissage est suffisamment précise ou si l'utilisation du modèle de simulation est requise. Cette décision doit être prise sans mesurer directement cette erreur de prédiction. Des contraintes sur le nombre d'instances de simulations pouvant être lancées en parallèle à tout instant doivent également être respectées. La stratégie proposée est, ainsi, inspirée de la théorie de l'apprentissage actif. Plus précisément, la décision de lancer une simulation pour un point de données spécifique n'est pas prise au hasard mais considère une mesure d'utilité associé aux prédictions du modèle d'apprentissage automatique pour chaque point de données.

Quatre variantes de la stratégie de couplage proposée, correspondant à quatre mesures d'incertitudes des prédictions, sont comparées à deux stratégies de références sur huit jeux de données de la littérature pour des résultats encourageant. Le modèle de substitution utilisé au cours de ces expériences est un modèle de forêt aléatoires. La prédiction de ce modèle est la moyenne des prédictions d'arbres de décision individuels. Les expériences numériques réalisées démontrent, que, en particulier, les stratégies de couplage utilisant comme mesure d'incertitude des estimateurs de la variance des différents arbres permettent de réduire significativement l'erreur préquentielle du couple. Les erreurs obtenues avec ces stratégies sont, en particulier, plus faible que celle obtenues pour la stratégie de référence maximisant naïvement l'usage du modèle de simulation.

L'inclusion de méthodes de détection de dérives conceptuelles dans la stratégie proposée est également discutée. Quatre méthodes de détection de dérives sont évaluées en conjonction avec la stratégie proposée dans le cas particulier d'une dérive réelle abrupte. Ainsi, les expériences numériques réalisées ont permis de mettre en évidence une première limite de la stratégie proposée : le biais d'échantillonnage induit par la stratégie proposée peut, sur certains jeux de données, ralentir la détection de dérives. Ce problème est observé pour toutes les méthodes de détection de dérives évaluées.

Les troisième et quatrième chapitres de ce mémoire proposent une application au cas des scieries, et plus précisément aux simulateurs de sciages. Tout d'abord, le troisième chapitre présente le résultat d'une revue de la littérature relative à la planification et au contrôle de la production en scierie. Il est montré que les modèles de simulation de sciage sont, en particulier, un outils très présent dans les solutions proposées par la littérature scientifique. L'utilisation de ces modèles de simulation est, en particulier, proposée pour générer des données de productions en l'absence d'historiques. Ces données servent ensuite de paramètres à des modèles de recherche opérationnelle pour l'optimisation de plans de production à moyen et long termes. L'absence de données historiques peut être, en effet, induite par l'introduction dans le plan de production de produits habituellement non présents dans le catalogue de la scierie, ou par un changement des paramètres des lignes de sciages pour favoriser la production de produits spécifiques. Il a, cependant, été noté que ces modèles de simulation peuvent être trop coûteux en calcul pour être utilisés pour des problèmes de décision opérationnel, à court terme. La résolution de ces problèmes

peut, en effet, requérir les résultats de plusieurs milliers de simulations, pour plusieurs centaines ou plusieurs milliers de grumes et plusieurs configurations de scieries. Il avait, cependant, été proposé dès 2015 de remplacer ces modèles de simulation par des modèles de substitution basés sur l'apprentissage automatique.

La stratégie de couplage introduite au deuxième chapitre est donc appliquée à ce cas d'usage à l'aide d'un jeu de données de résultats de simulation. Un nouveau modèle de substitution pour ces simulateurs de sciages est d'abord introduit. Les scans 3D des grumes de bois sont d'abord plongés dans un espace vectoriel par la méthode dite de plongement dans un sous-espace de dissimilarités. Chaque scan est ainsi, représenté par un vecteur dont chaque composante mesure sa dissimilarité à une certaine grume représentative. Cette représentation structurée est ensuite utilisée pour l'entraînement de modèles de forêts aléatoires, sélectionnées en particulier pour leurs bonnes performances lors de travaux précédents. Parmi les différentes dissimilarités évaluées pour construire le sous-espace de dissimilarité, les meilleurs résultats ont, ici, été obtenus avec la dissimilarité ICP. Cette dissimilarité est une conséquence de l'algorithme Iterative Closest Point.

Dans un deuxième temps, les variantes de la stratégie de couplage basées sur les quatre mesures d'incertitude définies précédemment sont évaluées et comparées aux deux mesures de référence. Il est montré que la mesure d'incertitude permettant la plus grande réduction de l'erreur préquentielle du couple est, une fois encore, la variance des prédictions des différents arbres de décision de la forêt aléatoire.

De plus, les quatre méthodes de détections de dérives précédemment évaluées uniquement dans le cas de la dérive réelle sont à nouveau évaluées, sur le jeu de données de résultats de simulation, à la fois dans les cas de dérives réelles et virtuelles. Il est montré que les méthodes de détection de dérives sont, sur ce jeu de données, sensibles au biais d'échantillonnage induit par la stratégie de couplage. Dans le cas de la dérive réelle, ce biais a un impact négatif sur le temps de détection de dérives. Cependant, dans le cas de la dérive virtuelle, le fait que la stratégie de couplage est capable, même après l'instant de dérive, de détecter les points de données pour lesquels l'erreur de prédiction est importante induit une réduction du temps de détection.

Le chapitre 6 explicite les limites de la stratégie de couplage présentée dans ce mémoire et liste des pistes de travaux futurs. En particulier, l'échantillonnage actif du flux de données complique, en pratique l'évaluation et la comparaison de modèles de substitution qui pourraient être entraînés sur ces bases. De plus, certaines hypothèses de travail, comme l'indépendance entre les temps de simulation et l'erreur commise par le modèle de substitution pourraient ne pas être respectées en pratique. Plusieurs pistes de travaux futurs concernant l'amélioration de la méthode de couplage, son étude théorique et son intégration à un système réel sont également proposées.

Publications scientifiques

0.1 Revues internationales à comité de lecture

1. Sylvain Chabanet, Hind Bril El-Haouzi, and Philippe Thomas. “Coupling digital simulation and machine learning metamodel through an active learning approach in Industry 4.0 context”. In : *Computers in Industry* 133 (2021), p. 103529. (IF : 11.245)
2. Sylvain Chabanet et al. “Toward digital twins for sawmill production planning and control : benefits, opportunities, and challenges”. In : *International Journal of Production Research* (2022), pp. 1–24 (IF : 9.018 (2021))
3. Chabanet, S., Thomas, P., and Bril El-Haouzi, H. (2023). MLP Based on Dissimilarity Features : An Application to Wood Sawing Simulator Metamodeling. SN Computer Science, 4(4), 408.

0.2 Congrès internationaux à comité de lecture

1. Sylvain Chabanet, Philippe Thomas, and Hind Bril El-Haouzi. “Medoid-based MLP : an application to wood sawing simulator metamodeling”. In : *13th International Conference on Neural Computation Theory and Applications, NCTA 2021.* 2021.
2. Sylvain Chabanet et al. A knn approach based on icp metrics for 3d scans matching : an application to the sawing process”. In : *IFAC-PapersOnLine* 54.1 (2021), pp. 396–401.
3. Sylvain Chabanet et al. “Dissimilarity to class medoids as features for 3D point cloud classification”. In : *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems : IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part III.* Springer. 2021, pp. 573–581.
4. Sylvain Chabanet et al. "A comparison of wood log dissimilarities to predict sawmill output with k-Nearest Neighbor algorithms". In : *4th International Conference on Advances in Signal Processing and Artificial Intelligence, ASPAI’ 2022.* Oct 2022, Corfu, Greece.
5. Sylvain Chabanet, Philippe Thomas, and Hind Bril El-Haouzi. “Medoid-based MLP : an application to wood sawing simulator metamodeling”. In : *IFAC-PapersOnLine* 55.2 (2022), pp. 378–383.
6. Sylvain Chabanet, Hind Bril El Haouzi, and Philippe Thomas. “Toward a sawmill digital shadow based on coupled simulation and supervised learning models”. In : *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future : Proceedings of SOHOMA 2022.* Springer, 2023, pp. 59–70.

7. Sylvain Chabanet et al. "An object-oriented architecture to couple simulators and their machine learning surrogates models in the context of digital shadows" Accepted for presentation in *22nd World Congress of the International Federation of Automatic Control, IFAC World Congress* 2023, Yokohama, Japan.
8. 8. Sylvain Chabanet, Philippe Thomas, and Hind Bril El-Haouzi. "Active learning-based online coupling of sawmill simulators and their surrogate model : the effect of sampling bias on concept drift detection". Accepted for presentation in : *5th International Conference on Advances in Signal Processing and Artificial Intelligence*, ASPAI 2023, Tenerife, Spain.

Bibliographie

Abraham, Alexandre et Léo Dreyfus-Schmidt (2022). « Cardinal, a metric-based Active learning framework ». In : *Software Impacts* 12, p. 100250.

Alvarez, Pamela P et Jorge R Vera (2014). « Application of robust optimization to the sawmill planning problem ». In : *Annals of Operations Research* 219.1, p. 457-475.

Alvarez, Pamela P et al. (2020). « Improving consistency in hierarchical tactical and operational planning using Robust Optimization ». In : *Computers & Industrial Engineering* 139, p. 106112.

Amrouss, Amine et al. (2017). « Real-time management of transportation disruptions in forestry ». In : *Computers & Operations Research* 83, p. 95-105.

Angluin, Dana (1988). « Queries and concept learning ». In : *Machine learning* 2.4, p. 319-342.

Arlot, Sylvain (2018). *Validation croisée*. Sous la dir. de Myriam Maumy-Bertrand, Gilbert Saporta et Christine Thomas-Agnan. Dernier accès en Mai 2023. url : `https://hal.science/hal-01485508/file/hal_JES_validation-croisee.pdf`.

Arzamasov, Vadim, Klemens Böhm et Patrick Jochem (2018). « Towards concise models of grid stability ». In : *2018 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm)*. IEEE, p. 1-6.

Asadi, Mahyar et al. (2021). « Machine-learning-enabled digital twin of welded structures for rapid weld sequence design ». In : *74th IIW on-line Assembly and International Conference*.

Azangoo, Mohammad et al. (2021). « Hybrid Digital Twin for process industry using Apros simulation environment ». In : *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, p. 01-04.

Baesler, Felipe et Cristian Palma (2014). « Multiobjective parallel machine scheduling in the sawmill industry using memetic algorithms ». In : *The International Journal of Advanced Manufacturing Technology* 74.5, p. 757-768.

Baier, Lucas et al. (2021). « Detecting concept drift with neural network model uncertainty ». In : *arXiv preprint arXiv :2107.01873*.

Bajgiran, Omid Sanei, Masoumeh Kazemi Zanjani et Mustapha Nourelfath (2016). « The value of integrated tactical planning optimization in the lumber supply chain ». In : *International Journal of Production Economics* 171, p. 22-33.

Baram, Yoram, Ran El Yaniv et Kobi Luz (2004). « Online choice of active learning algorithms ». In : *Journal of Machine Learning Research* 5.Mar, p. 255-291.

Beaudoin, Daniel, J-M Frayret et Luc LeBel (2010). « Negotiation-based distributed wood procurement planning within a multi-firm environment ». In : *Forest Policy and Economics* 12.2, p. 79-93.

Beaudoin, Daniel, Luc LeBel et Mohamed Amine Soussi (2012). « Discrete event simulation to improve log yard operations ». In : *INFOR : Information Systems and Operational Research* 50.4, p. 175-185.

Bermeo-Ayerbe, Miguel Angel, Carlos Ocampo-Martinez et Javier Diaz-Rozo (2022). « Data-driven energy prediction modeling for both energy efficiency and maintenance in smart manufacturing systems ». In : *Energy* 238, p. 121691.

Besl, P. J. et N. D. McKay (1992). « A method for registration of 3-D shapes ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2, p. 239-256.

Boschert, Stefan et Roland Rosen (2016). « Digital twin—the simulation aspect ». In : *Mechatronic futures*. Springer, p. 59-74.

Bottani, Eleonora et al. (2017). « From the Cyber-Physical System to the Digital Twin : the process development for behaviour modelling of a Cyber Guided Vehicle in M2M logic ». In : *XXII Summer School Francesco TurcoIndustrial Systems Engineering*, p. 1-7.

Boukherroub, Tasseda et al. (2013a). *An Integrated Approach for Sustainable Supply Chain Planning : The Case of Divergent Processes*. Dernier accès en Mai 2023. url : `https://www.cirrelt.ca/documentstravail/cirrelt-fsa-2013-63.pdf`.

– (2013b). « An integrated approach for the optimization of the sustainable performance : A wood supply chain ». In : *IFAC Proceedings Volumes* 46.9, p. 186-191.

– (2015). « An integrated approach for sustainable supply chain planning ». In : *Computers & Operations Research* 54, p. 180-194.

Breiman, Leo (2001). « Random forests ». In : *Machine learning* 45.1, p. 5-32.

Bril El Haouzi, Hind (2017). « Contribution à la conception et à l'évaluation des architectures de pilotage des systèmes de production adaptables : vers une approche anthropocentrée pour la simulation et le pilotage ». dernier accès en Juin 2023. Habilitation à diriger des recherches. url : `https://hal.science/tel-01875639`.

Burbidge, Robert, Jem J Rowland et Ross D King (2007). « Active learning for regression based on query by committee ». In : *International conference on intelligent data engineering and automated learning*. Springer, p. 209-218.

Butt, Javaid et Vahaj Mohaghegh (2022). « Combining Digital Twin and Machine Learning for the Fused Filament Fabrication Process ». In : *Metals* 13.1, p. 24.

Caballero, Rafael et al. (2009). « Sawing planning using a multicriteria approach ». In : *Journal of Industrial & Management Optimization* 5.2, p. 303.

Candanedo, Luis M, Véronique Feldheim et Dominique Deramaix (2017). « Data driven prediction models of energy use of appliances in a low-energy house ». In : *Energy and buildings* 140, p. 81-97.

Cardin, Olivier et Damien Trentesaux (2022). « Design and Use of Human Operator Digital Twins in Industrial Cyber-Physical Systems : Ethical Implications ». In : *IFAC-PapersOnLine* 55.2, p. 360-365.

Chabanet, Sylvain, Hind Bril El-Haouzi et Philippe Thomas (2021). « Coupling digital simulation and machine learning metamodel through an active learning approach in Industry 4.0 context ». In : *Computers in Industry* 133, p. 103529.

– (2022). « Toward a self-adaptive digital twin based Active learning method : an application to the lumber industry ». In : *IFAC-PapersOnLine* 55.2, p. 378-383.

– (2023). « Toward a sawmill digital shadow based on coupled simulation and supervised learning models ». In : *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future : Proceedings of SOHOMA 2022*. Springer, p. 59-70.

Chabanet, Sylvain, Philippe Thomas et Hind Bril El-Haouzi (2021). « Medoid-based MLP : an application to wood sawing simulator metamodeling ». In : *13th International Conference on Neural Computation Theory and Applications, NCTA 2021*.

Chabanet, Sylvain et al. (2021). « A kNN approach based on ICP metrics for 3D scans matching : an application to the sawing process ». In : *17th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2021*.

Chabanet, Sylvain et al. (2023). « An object-oriented architecture to couple simulators and their machine learning surrogates models in the context of digital shadows ». In : *IFAC World Congress*. Accepté pour présentation.

Chalayer, Maurice (2017). *Le futur de la scierie française*. l'Harmattan.

Chopard, Bastien et Marco Tomassini (2018). *An introduction to metaheuristics for optimization*. Springer.

Cid, F et al. (2007). « Evaluatión of pull strategies in lumber production planninig : A case study ». In : *19th International Conference on Production Research*. Citeseer.

Cid Yáñez, Fabian et al. (2009). « Agent-based simulation and analysis of demand-driven production strategies in the timber industry ». In : *International journal of production research* 47.22, p. 6295-6319.

Cortez, Paulo et al. (2009). « Modeling wine preferences by data mining from physicochemical properties ». In : *Decision support systems* 47.4, p. 547-553.

Cybenko, George (1989). « Approximation by superpositions of a sigmoidal function ». In : *Mathematics of control, signals and systems* 2.4, p. 303-314.

D'Amours, Sophie, Mikael Rönnqvist et Andres Weintraub (2008). « Using operational research for supply chain planning in the forest products industry ». In : *INFOR : Information Systems and Operational Research* 46.4, p. 265-281.

Dasari, Siva Krishna, Abbas Cheddad et Petter Andersson (2019). « Random forest surrogate models to support design space exploration in aerospace use-case ». In : *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, p. 532-544.

Dasgupta, Sanjoy (2011). « Two faces of active learning ». In : *Theoretical computer science* 412.19, p. 1767-1781.

Dems, Amira, Louis-Martin Rousseau et Jean-Marc Frayret (2015). « Effects of different cut-to-length harvesting structures on the economic value of a wood procurement planning problem ». In : *Annals of Operations Research* 232.1, p. 65-86.

Domingos, Pedro et Geoff Hulten (2000). « Mining high-speed data streams ». In : *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 71-80.

Donald, W Stuart, Thomas C Maness et Marian V Marinescu (2001). « Production planning for integrated primary and secondary lumber manufacturing ». In : *Wood and Fiber Science* 33.3, p. 334-344.

Duin, RP et E Pękalska (2009). *The dissimilarity representation for pattern recognition : a tutorial*. Dernier accès en Mai 2023. url : `https://rduin.nl/presentations/DisRep_Tutorial_doc.pdf`.

Dumetz, Ludwig et al. (2015). « A simulation framework for the evaluation of production planning and order management strategies in the sawmilling industry ». In : *IFAC-PapersOnLine* 48.3, p. 622-627.

Dumetz, Ludwig et al. (2017). *Planning and coordination for decentralised business units in a lumber production company*. Dernier accès en Mai 2023. url : `https://hal.science/hal-01425811`.

Dumetz, Ludwig et al. (2021). « Tactical-operational coordination of a divergent production system with coproduction : the sawmilling challenge ». In : *INFOR : Information Systems and Operational Research*, p. 1-23.

Dumont, Laurence B et al. (2019). « Integrating Electric Energy Cost in Lumber Production Planning ». In : *IFAC-PapersOnLine* 52.13, p. 2249-2254.

Dyer, M.E et A.M Frieze (1985). « A simple heuristic for the p-centre problem ». In : *Operations Research Letters* 3.6, p. 285-288.

Farache, David E et al. (2022). « Active learning and molecular dynamics simulations to find high melting temperature alloys ». In : *Computational Materials Science* 209, p. 111386.
Fletcher, Martyn et al. (2001). « Reconfiguring processes in a holonic sawmill ». In : *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*. T. 1. IEEE, p. 158-163.
Forget, Pascal, Sophie D'Amours et Jean-Marc Frayret (2008). « Multi-behavior agent model for planning in supply chains : An application to the lumber industry ». In : *Robotics and Computer-Integrated Manufacturing* 24.5, p. 664-679.
Forget, Pascal et al. (2009). « Study of the performance of multi-behaviour agents for supply chain planning ». In : *Computers in industry* 60.9, p. 698-708.
Frayret, J-M et al. (2007). « Agent-based supply-chain planning in the forest products industry ». In : *International Journal of Flexible Manufacturing Systems* 19.4, p. 358-391.
Gama, Joao, Ricardo Rocha et Pedro Medas (2003). « Accurate decision trees for mining high-speed data streams ». In : *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 523-528.
Gama, Joao, Raquel Sebastiao et Pedro Pereira Rodrigues (2013). « On evaluating stream learning algorithms ». In : *Machine learning* 90.3, p. 317-346.
Gao, Mingfei et al. (2020). « Consistency-based semi-supervised active learning : Towards minimizing labeling cost ». In : *European Conference on Computer Vision*. Springer, p. 510-526.
Gardner, Paul et al. (2020). « Towards the development of an operational digital twin ». In : *Vibration* 3.3, p. 235-265.
Gaudreault, Jonathan, Jean-Marc Frayret et Gilles Pesant (2009). « Distributed search for supply chain coordination ». In : *Computers in Industry* 60.6, p. 441-451.
Gaudreault, Jonathan et al. (2010). « Distributed operations planning in the lumber supply chain : Models and coordination ». In : *International Journal of Industrial Engineering : Theory, Applications and Practice* 17.3, p. 168-189.
Gaudreault, Jonathan et al. (2011). « Combined planning and scheduling in a divergent production system with co-production : A case study in the lumber industry ». In : *Computers & Operations Research* 38.9, p. 1238-1250.
Gaudreault, Jonathan et al. (2012). « Supply chain coordination using an adaptive distributed search strategy ». In : *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6, p. 1424-1438.
Gomes, Heitor Murilo et al. (2019). « Machine learning for streaming data : state of the art, challenges, and opportunities ». In : *ACM SIGKDD Explorations Newsletter* 21.2, p. 6-22.
Goulet, Pierre (2006). *Optitek : User's manual*.
Grieves, Michael (2014). « Digital twin : manufacturing excellence through virtual factory replication ». In : *White paper* 1.2014, p. 1-7.
Grieves, Michael et John Vickers (2017). « Digital twin : Mitigating unpredictable, undesirable emergent behavior in complex systems ». In : *Transdisciplinary perspectives on complex systems*. Springer, p. 85-113.
Haag, Sebastian et Reiner Anderl (2018). « Digital twin–Proof of concept ». In : *Manufacturing letters* 15, p. 64-66.
Haberl, Josef et al. (1991). « A branch-and-bound algorithm for solving a fixed charge problem in the profit optimization of sawn timber production ». In : *Zeitschrift für Operations Research* 35.2, p. 151-166.
Hamidieh, Kam (2018). « A data-driven statistical model for predicting the critical temperature of a superconductor ». In : *Computational Materials Science* 154, p. 346-354.

He, Li, Xiaolong Wang et Hong Zhang (2016). « M2DP : A novel 3D point cloud descriptor and its application in loop closure detection ». In : *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, p. 231-237.

Hinkley, David V (1971). « Inference about the change-point from cumulative sum tests ». In : *Biometrika* 58.3, p. 509-523.

Hughes, Aidan J et al. (2022). « On robust risk-based active-learning algorithms for enhanced decision support ». In : *Mechanical Systems and Signal Processing* 181, p. 109502.

Huka, Maria Anna et Manfred Gronalt (2017). « Model development and comparison of different heuristics for production planning in large volume softwood sawmills ». In : *Engineering Optimization* 49.11, p. 1829-1847.

Huka, Maria Anna, Christian Rindler et Manfred Gronalt (2020). « Scheduling and loading problem for multiple, identical dry kilns ». In : *Flexible Services and Manufacturing Journal*, p. 1-25.

Hürkamp, André et al. (2021). « Machine learning and simulation-based surrogate modeling for improved process chain operation ». In : *The International Journal of Advanced Manufacturing Technology* 117.7, p. 2297-2307.

Ienco, Dino et al. (2013). « Clustering based active learning for evolving data streams ». In : *International Conference on Discovery Science*. Springer, p. 79-93.

Ismail Fawaz, Hassan et al. (2019). « Deep learning for time series classification : a review ». In : *Data Mining and Knowledge Discovery* 33.4, p. 917-963.

Jover, Jeremy, André Thomas et Vincent Bombardier (2011). « Marquage du bois dans la masse : Intérêts et perspectives ». In : *9e Congrès International de Génie Industriel, CIGI 2011*, CDROM.

Jover, Jeremy et al. (2010). « Pertinence of new communicating material paradigm : A first step towards wood mass marking ». In : *New Achievements in Material and Environmental Science–Names' 10*.

Julien, Nathalie et Mohammed Adel Hamzaoui (2023). « Integrating Lean Data and Digital Sobriety in Digital Twins Through Dynamic Accuracy Management ». In : *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future : Proceedings of SOHOMA 2022*. Springer, p. 107-117.

Julien, Nathalie et Éric Martin (2020). *Le jumeau numérique : De l'intelligence artificielle à l'industrie agile*. Dunod.

Kaltenbrunner, Matthias, Maria Anna Huka et Manfred Gronalt (2020). « Adaptive Model Building Framework for Production Planning in the Primary Wood Industry ». In : *Forests* 11.12, p. 1256.

Kar, Purushottam et Prateek Jain (2011). « Similarity-based learning via data driven embeddings ». In : *Advances in neural information processing systems* 24.

Kaya, Heysem, Pinar Tüfekci et Erdinç Uzun (2019). « Predicting CO and NOx emissions from gas turbines : novel data and a benchmark PEMS ». In : *Turkish Journal of Electrical Engineering and Computer Sciences* 27.6, p. 4783-4796.

Kazemi Zanjani, M, Mustapha Nourelfath et Daoud Aït-Kadi (2011). « Production planning with uncertainty in the quality of raw materials : a case in sawmills ». In : *Journal of the Operational Research Society* 62.7, p. 1334-1343.

Kazemi Zanjani, Masoumeh, Daoud Ait-Kadi et Mustapha Nourelfath (2013). « A stochastic programming approach for sawmill production planning ». In : *International Journal of Mathematics in Operational Research* 5.1, p. 1-18.

Kazemi Zanjani, Masoumeh, Mustapha Nourelfath et Daoud Ait-Kadi (2010). « A multi-stage stochastic programming approach for production planning with uncertainty in the quality

of raw materials and demand ». In : *International Journal of Production Research* 48.16, p. 4701-4723.

Kogler, Christoph et Peter Rauch (2020). « Contingency plans for the wood supply chain based on bottleneck and queuing time analyses of a discrete event simulation ». In : *Forests* 11.4, p. 396.

Kottke, Daniel, Georg Krempl et Myra Spiliopoulou (2015). « Probabilistic active learning in datastreams ». In : *International Symposium on Intelligent Data Analysis*. Springer, p. 145-157.

Kritzinger, Werner et al. (2018). « Digital Twin in manufacturing : A categorical literature review and classification ». In : *IFAC-PapersOnLine* 51.11, p. 1016-1022.

Kumar, Punit et Atul Gupta (2022). « Active instance selection via parametric equation and instance overlap aware scheme ». In : *Applied Intelligence* 52.1, p. 994-1012.

Ladj, Asma et al. (2021). « A knowledge-based Digital Shadow for machining industry in a Digital Twin perspective ». In : *Journal of Manufacturing Systems* 58, p. 168-179.

Laguna, Manuel (2018). « Tabu Search ». In : *Handbook of Heuristics*. Sous la dir. de Rafael Martí, Panos M. Pardalos et Mauricio G. C. Resende. Cham : Springer International Publishing, p. 741-758.

Lakshminarayanan, Balaji, Daniel M Roy et Yee Whye Teh (2014). « Mondrian forests : Efficient online random forests ». In : *Advances in neural information processing systems* 27.

Lemieux, Sébastien et al. (2009). « Agent-based simulation to anticipate impacts of tactical supply chain decision-making in the lumber industry ». In : *International Journal of Flexible Manufacturing Systems* 19.4, p. 358-391.

Lindner, Berndt, Tanya Visser et Brand Wessels (2013). « A model to optimise the linked sawing and ripping decisions in the South African pine wood industry ». In : *42th Annual Conference of the Operations Research Society of South Africa (ORSSA). Protea Hotel Technopark, Stellenbosch, South Africa*.

Lindner, Berndt G, PJ Vlok et C Brand Wessels (2015). « Determining optimal primary sawing and ripping machine settings in the wood manufacturing chain ». In : *Southern Forests : a Journal of Forest Science* 77.3, p. 191-201.

Little, John DC et Stephen C Graves (2008). « Little's law ». In : *Building intuition : insights from basic operations management models and principles*, p. 81-100.

Liu, Mengnan et al. (2021). « Review of digital twin about concepts, technologies, and industrial applications ». In : *Journal of Manufacturing Systems* 58, p. 346-361.

Lobos, Alfonso et Jorge R Vera (2016). « Intertemporal stochastic sawmill planning : Modeling and managerial insights ». In : *Computers & Industrial Engineering* 95, p. 53-63.

Loghin, Adrian et Shakhrukh Ismonov (2022). « 3D FEA based surrogate modeling in fatigue crack growth life assessment ». In : *Procedia Structural Integrity* 38, p. 331-341.

Losing, Viktor, Barbara Hammer et Heiko Wersing (2016). « KNN classifier with self adjusting memory for heterogeneous concept drift ». In : *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, p. 291-300.

Lu, Zhou et al. (2017). « The expressive power of neural networks : A view from the width ». In : *Advances in neural information processing systems* 30.

Luo, Congnan, Yanjun Li et Soon M Chung (2009). « Text document clustering based on neighbors ». In : *Data & Knowledge Engineering* 68.11, p. 1271-1288.

Maness, Thomas C et Scott E Norton (2002). « Multiple period combined optimization approach to forest production planning ». In : *Scandinavian Journal of Forest Research* 17.5, p. 460-471.

Marier, Philippe, Jonathan Gaudreault et Thomas Noguer (2021). « Kiln drying operations scheduling with dynamic composition of loading patterns ». In : *Forest Products Journal* 71.2, p. 101-110.

Marier, Philippe, Jonathan Gaudreault et Benoit Robichaud (2014). « Implementing a MIP model to plan and schedule wood finishing operations in a sawmill : lessons learned ». In : *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation.*

Marier, Philippe et al. (2014). « S&OP network model for commodity lumber products ». In : *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation.*

Marinescu, Marian V et Thomas C Maness (2010). « A hierarchical timber allocation model to analyze sustainable forest management decisions. » In : *Mathematical & Computational Forestry & Natural Resource Sciences* 2.2.

Martineau, Vincent et al. (2021). « Neural network architectures and feature extraction for lumber production prediction ». In : *The 34th Canadian Conference on Artificial Intelligence.*

Masood, Tariq et Paul Sonntag (2020). « Industry 4.0 : Adoption challenges and benefits for SMEs ». In : *Computers in Industry* 121, p. 103261.

Maturana, Sergio, Enzo Pizani et Jorge Vera (2010). « Scheduling production for a sawmill : A comparison of a mathematical model versus a heuristic ». In : *Computers & Industrial Engineering* 59.4, p. 667-674.

Melesse, Tsega Y., Valentina Di Pasquale et Stefano Riemma (2020). « Digital Twin Models in Industrial Operations : A Systematic Literature Review ». In : *Procedia Manufacturing* 42. International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019), p. 267-272.

Mendoza, GA et al. (1991). « Combining simulation and optimization models for hardwood lumber production ». In : *Proceedings, Pacific Rim Forestry–Bridging the World. pp. 356-361.*

Molina, Julian et al. (2007). « SSPMO : A scatter tabu search procedure for non-linear multiobjective optimization ». In : *INFORMS Journal on Computing* 19.1, p. 91-100.

Morin, Michael et al. (2015). « Machine learning-based metamodels for sawing simulation ». In : *2015 Winter Simulation Conference (WSC).* IEEE, p. 2160-2171.

Morin, Michael et al. (2020). « Machine learning-based models of sawmills for better wood allocation planning ». In : *International Journal of Production Economics* 222, p. 107508.

Morneau-Pereira, Maxime et al. (2014). « An optimization and simulation framework for integrated tactical planning of wood harvesting operations, wood allocation and lumber production ». In : *MOSIM 2014, 10eme Conférence Francophone de Modélisation, Optimisation et Simulation.*

Müller, Fabian, Dirk Jaeger et Marc Hanewinkel (2019). « Digitization in wood supply–A review on how Industry 4.0 will change the forest value chain ». In : *Computers and Electronics in Agriculture* 162, p. 206-218.

Müller, Meinard (2007). « Dynamic time warping ». In : *Information retrieval for music and motion*, p. 69-84.

Nguyen, Derrick et Bernard Widrow (1990). « Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights ». In : *1990 IJCNN International Joint Conference on Neural Networks.* IEEE, p. 21-26.

Nguyen, Van-Tho et al. (2020). « A machine-learning approach for classifying defects on tree trunks using terrestrial LiDAR ». In : *Computers and Electronics in Agriculture* 171, p. 105332.

Noyel, Mélanie et al. (2013). « Implantation of an on-line quality process monitoring ». In : *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM).* IEEE, p. 1-6.

Noyel, Melanie et al. (2016). « Reconfiguration process for neuronal classification models : Application to a quality monitoring problem ». In : *Computers in Industry* 83, p. 78-91.

Opacic, Luke, Taraneh Sowlati et Mahdi Mobini (2018). « Design and development of a simulation-based decision support tool to improve the production process at an engineered wood products mill ». In : *International journal of production economics* 199, p. 209-219.

Osada, Robert et al. (2001). « Matching 3D models with shape distributions ». In : *Proceedings international conference on shape modeling and applications*. IEEE, p. 154-166.
Page, Ewan S (1954). « Continuous inspection schemes ». In : *Biometrika* 41.1/2, p. 100-115.
Palma, Cristian D et Francisco P Vergara (2016). « A multiobjective model for the cutting pattern problem with unclear preferences ». In : *Forest Science* 62.2, p. 220-226.
Pękalska, Elżbieta, Robert PW Duin et Pavel Paclík (2006). « Prototype selection for dissimilarity-based classifiers ». In : *Pattern Recognition* 39.2, p. 189-208.
Pradenas, Lorena, Fernando Peñailillo et Jacques Ferland (2004). « Aggregate production planning problem. A new algorithm ». In : *Electronic Notes in Discrete Mathematics* 18, p. 193-199.
Pradenas, Lorena et al. (2013). « Genotype–phenotype heuristic approaches for a cutting stock problem with circular patterns ». In : *Engineering applications of Artificial Intelligence* 26.10, p. 2349-2355.
Pylianidis, Christos et al. (2022). « Simulation-assisted machine learning for operational digital twins ». In : *Environmental Modelling & Software* 148, p. 105274.
Qi, Charles R et al. (2017). « Pointnet : Deep learning on point sets for 3d classification and segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 652-660.
Rathke, Jörn, Maria A Huka, Manfred Gronalt et al. (2013). « The box assignment problem in log yards ». In : *Silva Fennica* 47.3, p. 1-13.
Razali, Nornadiah Mohd, Yap Bee Wah et al. (2011). « Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests ». In : *Journal of statistical modeling and analytics* 2.1, p. 21-33.
Razavi, Saman, Bryan A Tolson et Donald H Burn (2012). « Review of surrogate modeling in water resources ». In : *Water Resources Research* 48.7.
Read, Jesse et al. (2012). « Batch-incremental versus instance-incremental learning in dynamic and evolving data ». In : *International symposium on intelligent data analysis*. Springer, p. 313-323.
Reis, Denis Moreira dos et al. (2016). « Fast unsupervised online drift detection using incremental kolmogorov-smirnov test ». In : *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1545-1554.
Reitz, Jan, Michael Schluse et Jürgen Rossmann (2019). « Industry 4.0 beyond the factory : An application to forestry ». In : *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*. Springer, p. 107-116.
Ríos, José et al. (2015). « Product avatar as digital counterpart of a physical individual product : Literature review and implications in an aircraft ». In : *Transdisciplinary Lifecycle Analysis of Systems*, p. 657-666.
Ritto, TG et FA Rochinha (2021). « Digital twin, physics-based model, and machine learning applied to damage detection in structures ». In : *Mechanical Systems and Signal Processing* 155, p. 107614.
Rönnqvist, Mikael (2003). « Optimization in forestry ». In : *Mathematical programming* 97.1, p. 267-284.
Rönnqvist, Mikael et al. (2015). « Operations research challenges in forestry : 33 open problems ». In : *Annals of Operations Research* 232.1, p. 11-40.
Ross, Gordon J et al. (2012). « Exponentially weighted moving average charts for detecting concept drift ». In : *Pattern recognition letters* 33.2, p. 191-198.
Santa-Eulalia, Luis Antonio et al. (2011). « Agent-based experimental investigations of the robustness of tactical planning and control policies in a softwood lumber supply chain ». In : *Production planning & control* 22.8, p. 782-799.

Santos, Carlos Henrique dos et al. (2021). « Decision support in productive processes through DES and ABS in the Digital Twin era : a systematic literature review ». In : *International Journal of Production Research*, p. 1-20.
Sapel, Patrick et al. (2022). « Towards digital shadows for production planning and control in injection molding ». In : *CIRP Journal of Manufacturing Science and Technology* 38, p. 243-251.
Savadjiev, Peter et al. (2020). « Knowledge based versus data based : A historical perspective on a continuum of methodologies for medical image analysis ». In : *Neuroimaging Clinics* 30.4, p. 401-415.
Savolainen, Jyrki et Mikkel Stein Knudsen (2022). « Contrasting digital twin vision of manufacturing with the industrial reality ». In : *International Journal of Computer Integrated Manufacturing* 35.2, p. 165-182.
Schleif, Frank-Michael et Peter Tino (2015). « Indefinite proximity learning : A review ». In : *Neural Computation* 27.10, p. 2039-2096.
Scholz, Fritz W et Michael A Stephens (1987). « K-sample Anderson–Darling tests ». In : *Journal of the American Statistical Association* 82.399, p. 918-924.
Schroeder, Greyce N et al. (2016). « Digital twin data modeling with automationml and a communication methodology for data exchange ». In : *IFAC-PapersOnLine* 49.30, p. 12-17.
Selma, Cyrine et al. (2018). « An iterative closest point method for measuring the level of similarity of 3D log scans in wood industry ». In : *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, p. 433-444.
Semeraro, Concetta et al. (2021). « Digital twin paradigm : A systematic literature review ». In : *Computers in Industry* 130, p. 103469.
– (2022). « Data-driven Invariant Modelling Patterns for Digital Twin Design ». In : *Journal of Industrial Information Integration*, p. 100424.
Settles, Burr (2009). *Active learning literature survey*. Dernier accès en Mai 2023. url : `https://minds.wisconsin.edu/handle/1793/60660`.
Shafto, Mike et al. (2012). « Modeling, simulation, information technology & processing roadmap ». In : *National Aeronautics and Space Administration* 32.2012, p. 1-38.
Shahhosseini, Mohsen et al. (2021). « Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt ». In : *Scientific reports* 11.1, p. 1-15.
Shahi, Shashi et Reino Pulkki (2015). « A simulation-based optimization approach to integrated inventory management of a sawlog supply chain with demand uncertainty ». In : *Canadian Journal of Forest Research* 45.10, p. 1313-1326.
Shahi, Shashi Kamal (2016). « Supply chain management of the Canadian Forest Products industry under supply and demand uncertainties : a simulation-based optimization approach ». Thèse de doct. Faculty of Natural Resources Management.
Singh, Maulshree et al. (2021). « Digital twin : Origin to future ». In : *Applied System Innovation* 4.2, p. 36.
Skog, Johan, Mikael Jacobsson, Anders Lycken et al. (2017). « Traceability and adaptive production in the digital sawmill ». In : *Pro Ligno* 13.4, p. 162-167.
Smith, Temple F, Michael S Waterman et al. (1981). « Identification of common molecular subsequences ». In : *Journal of molecular biology* 147.1, p. 195-197.
Sohrabi, Pegah (2013). « A Three-stage Control Mechanism for the Lumber Production Process of a Sawmill Based on a Powers-of-two Modelling Approach ». Mém. de mast. Dalhousie University.
Spinti, Jennifer P, Philip J Smith et Sean T Smith (2022). « Atikokan Digital Twin : Machine learning in a biomass energy system ». In : *Applied Energy* 310, p. 118436.

Staudhammer, Christina, Thomas C Maness et Robert A Kozak (2007). « Profile charts for monitoring lumber manufacturing using laser range sensor data ». In : *Journal of Quality Technology* 39.3, p. 224-240.

Tan, Chang How, Vincent CS Lee et Mahsa Salehi (2022). « Information resources estimation for accurate distribution-based concept drift detection ». In : *Information Processing & Management* 59.3, p. 102911.

Tao, Fei et al. (2019). « Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0 : Correlation and comparison ». In : *Engineering* 5.4, p. 653-661.

Theresia, Juliet, I Gede Agus Widyadana et Didik Wahjudi (2019). « Optimal Kiln Dry Allocation for Dry Timber Preparation to Minimize Cost ». In : *Jurnal Teknik Industri* 21.1, p. 43-48.

Thomas, Philippe, Denise Choffel et André Thomas (2008). « Simulation reduction models approach using neural network ». In : *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*. IEEE, p. 679-684.

Thomas, Philippe et André Thomas (2011). « Multilayer perceptron for simulation models reduction : Application to a sawmill workshop ». In : *Engineering Applications of Artificial Intelligence* 24.4, p. 646-657.

Todoraki, C et Mikael Ronnqvist (2001). « Log sawing optimisation directed by market demand ». In : *New Zealand Journal of Forestry* 45, p. 29-33.

Troncoso, Juan et al. (2015). « A mixed integer programming model to evaluate integrating strategies in the forest value chain—a case study in the Chilean forest industry ». In : *Canadian Journal of Forest Research* 45.7, p. 937-949.

Troncoso, Juan J et Rodrigo A Garrido (2005). « Forestry production and logistics planning : an analysis using mixed-integer programming ». In : *Forest Policy and Economics* 7.4, p. 625-633.

Tuegel, Eric (2012). « The airframe digital twin : some challenges to realization ». In : *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, p. 1812.

Tüfekci, Pınar (2014). « Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods ». In : *International Journal of Electrical Power & Energy Systems* 60, p. 126-140.

Vahid, Saba (2011). « An agent-based supply chain model for strategic analysis in forestry ». Thèse de doct. University of British Columbia.

Vahidian, Naghmeh (2012). « Comparison of Deterministic and Stochastic Production Planning Approaches in Sawmills by Integrating Design of Experiments and Monte-Carlo simulation ». Thèse de doct. Concordia University Montreal, Quebec, Canada.

Vanzetti, Nicolás, Gabriela Corsano et Jorge M Montagna (2020). « Drying operation planning in a sawmill ». In : *Computers & Chemical Engineering* 137, p. 106817.

Vanzetti, Nicolás et al. (2018). « An optimization approach for multiperiod production planning in a sawmill ». In : *Forest Policy and Economics* 97, p. 1-8.

– (2019a). « A detailed mathematical programming model for the optimal daily planning of sawmills ». In : *Canadian Journal of Forest Research* 49.11, p. 1400-1411.

Vanzetti, Nicolás et al. (2019b). « Integrated approach for the bucking and production planning problems in forest industry ». In : *Computers & Chemical Engineering* 125, p. 155-163.

Varas, Mauricio et al. (2014). « Scheduling production for a sawmill : A robust optimization approach ». In : *International Journal of Production Economics* 150, p. 37-51.

Vergara, Francisco P, Cristian D Palma et Héctor Sepúlveda (2015). « A comparison of optimization models for lumber production planning ». In : *Bosque* 36.2, p. 239-246.

Vila, Didier, Robert Beauregard et Alain Martel (2009). « The strategic design of forest industry supply chains ». In : *INFOR : Information Systems and Operational Research* 47.3, p. 185-202.
Wager, Stefan, Trevor Hastie et Bradley Efron (2014). « Confidence intervals for random forests : The jackknife and the infinitesimal jackknife ». In : *The Journal of Machine Learning Research* 15.1, p. 1625-1651.
Wang, Jinjiang et al. (2019). « Digital Twin for rotating machinery fault diagnosis in smart manufacturing ». In : *International Journal of Production Research* 57.12, p. 3920-3934.
Wang, Jinjiang et al. (2022). « Hybrid physics-based and data-driven models for smart manufacturing : Modelling, simulation, and explainability ». In : *Journal of Manufacturing Systems* 63, p. 381-391.
Wang, Liantao et al. (2015). « Active learning via query synthesis and nearest neighbour search ». In : *Neurocomputing* 147, p. 426-434.
Wang, Min et al. (2017). « Active learning through density clustering ». In : *Expert systems with applications* 85, p. 305-317.
Webb, Geoffrey I et al. (2016). « Characterizing concept drift ». In : *Data Mining and Knowledge Discovery* 30.4, p. 964-994.
Weigl, Eva et al. (2016). « On improving performance of surface inspection systems by online active learning and flexible classifier updates ». In : *Machine Vision and Applications* 27.1, p. 103-127.
Wery, Jean et al. (2014). « Decision-making framework for tactical planning taking into account market opportunities (new products and new suppliers) in a co-production context ». In : *10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14*.
Wery, Jean et al. (2018). « Simulation-optimisation based framework for Sales and Operations Planning taking into account new products opportunities in a co-production context ». In : *Computers in industry* 94, p. 41-51.
Wohlin, Claes (2014). « Guidelines for snowballing in systematic literature studies and a replication in software engineering ». In : *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, p. 1-10.
Xu, Jie et al. (2015). « Simulation optimization : A review and exploration in the new era of cloud computing and big data ». In : *Asia-Pacific Journal of Operational Research* 32.03, p. 1550019.
Xu, Zuobing, Ram Akella et Yi Zhang (2007). « Incorporating diversity and density in active learning for relevance feedback ». In : *European Conference on Information Retrieval*. Springer, p. 246-257.
Yeh, I-C (1998). « Modeling of strength of high-performance concrete using artificial neural networks ». In : *Cement and Concrete research* 28.12, p. 1797-1808.
Yu, Jianhui et al. (2021). *3d medical point transformer : Introducing convolution to attention networks for medical point cloud analysis*. Dernier accès en Juin 2023. url : `https://arxiv.org/abs/2112.04863`.
Zanjani, M Kazemi, Mustapha Nourelfath et Daoud Ait-Kadi (2013a). « A scenario decomposition approach for stochastic production planning in sawmills ». In : *Journal of the Operational Research Society* 64.1, p. 48-59.
Zanjani, Masoumeh Kazemi, Daoud Ait-Kadi et Mustapha Nourelfath (2010). « Robust production planning in a manufacturing environment with random yield : A case in sawmill production planning ». In : *European Journal of Operational Research* 201.3, p. 882-891.
– (2013). « An accelerated scenario updating heuristic for stochastic production planning with set-up constraints in sawmills ». In : *International Journal of Production Research* 51.4, p. 993-1005.

Zanjani, Masoumeh Kazemi, Mustapha Nourelfath et Daoud Ait-Kadi (2007). *A stochastic programming approach for production planning in a manufacturing environment with random yield*. Dernier accès en Mai 2023. URL : https://epe.lac-bac.gc.ca/100/200/300/cirrelt/CIRRELT-2007-58/CIRRELT-2007-58.pdf.

– (2013b). « Sawmill Production Planning Under Uncertainty : Modelling and Solution Approaches ». In : *Stochastic Programming : Applications in Finance, Energy, Planning and Logistics*. World Scientific, p. 347-395.

Zhang, Hao et al. (2017). « A digital twin-based approach for designing and multi-objective optimization of hollow glass production line ». In : *Ieee Access* 5, p. 26901-26911.

Zheng, Xiaochen, Jinzhi Lu et Dimitris Kiritsis (2021). « The emergence of cognitive digital twin : vision, challenges and opportunities ». In : *International Journal of Production Research* 0.0, p. 1-23.

Zhou, Fang, Q Claire et Ross D King (2014). « Predicting the geographical origin of music ». In : *2014 IEEE International Conference on Data Mining*. IEEE, p. 1115-1120.

Zhou, Guanghui et al. (2020). « Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing ». In : *International Journal of Production Research* 58.4, p. 1034-1051.

Žliobaitė, Indrė et al. (2013). « Active learning with drifting streaming data ». In : *IEEE transactions on neural networks and learning systems* 25.1, p. 27-39.

Annexe A

Algorithme d'Anderson Darling Incremental

Cette annexe présente un algorithme permettant de mettre à jour de façon incrémentale la statistique du test d'Anderson-Darling pour comparer une fenêtre de données fixe et une fenêtre glissante.

Les hypothèses d'application de cette méthode sont les suivantes :

- Le test est réalisé entre deux échantillons S_1 et S_2.
- Les deux échantillons sont de taille fixe est égale à $n = \frac{N}{2}$. L'échantillon S_1 est fixé et ne change pas avec le temps.
- L'échantillon S_2 est constitué des données présentent dans une fenêtre glissante sur le flux. Lorsqu'une nouvelle donnée est générée par le flux, elle remplace la valeur la plus ancienne dans S_2.

Dans un cadre général, la statistique du test d'Anderson-Darling à k échantillons a été définie par Scholz et Stephens, 1987. En pratique, elle peut n'être qu'en fonction de rangs dans un échantillon ordonné :

$$A^2_{kN} = \frac{1}{N} \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{N-1} \frac{(NM_{ij} - jn_i)^2}{j(N-j)}\,, \tag{A.1}$$

avec n_i la taille de l'échantillon i, $N = \sum_i n_i$ la taille de l'échantillon total regroupant la totalité des observations, $Z_1 < ... < Z_j < ... < Z_N$ les données ordonnées dans l'échantillon total et M_{ij} le nombre d'observations dans l'échantillon i inférieures ou égales à Z_j.

Dans le cas simplifié à deux échantillons de même taille, $M_{1j} + M_{2j} = j$ pour tout j, la statistique A^2_{kN} peut être simplifiée en :

$$A^2_{2N} = 4 \sum_{j=1}^{N-1} \frac{(M_{1j} - \frac{j}{2})^2}{j(N-j)}. \tag{A.2}$$

$$= 4 \sum_{j=1}^{N-1} \frac{M^2_{1j}}{j(N-j)} - 4 \sum_{j=1}^{N-1} \frac{M_{1j}}{N-j} + \sum_{j=1}^{N-1} \frac{j}{N-j} \tag{A.3}$$

Introduisons à présent la table de données A.1 contenant les informations nécessaires pour calculer et incrémenter la statistique d'Anderson Darling.

Z_1		Z_j		Z_N
s_1		s_j		s_N
t_1	...	t_j	...	t_N
M_{11}		M_{1j}		M_{1N}

TABLE A.1 : Table nécessaire au calcul et à l'incrémentation de A^2_{2N}

Dans cette table, les t_j sont les temps d'arrivées des différents éléments dans le flux, utilisé pour déterminer qu'elle est la donnée la plus ancienne dans la fenêtre S_2. s_j est un booléen qui indique si la valeur Z_j appartient à l'échantillon S_1 ou à l'échantillon S_2. Le calcul de A^2_{2N} ne fait intervenir que les dernières lignes du tableau. Les autres lignes sont utilisées pour incrémenter la statistique lors de l'arrivée de nouvelles données.

Lorsqu'une nouvelle donnée arrive, celle-ci doit remplacer la plus ancienne valeur de l'échantillon S_2, suivant le concept de fenêtre glissante. Cette donnée peut être insérée dans la table A.1 comme suit :

- Identifier la colonne p correspondant à la valeur la plus ancienne de l'échantillon S_2.
- Remplacer la valeur Z_p et le temps d'arrivée associé t_p par leurs nouvelles valeurs.
- Échanger successivement la colonne p avec les colonnes de gauche ou de droite jusqu'à ce que les (Z_j) soit de nouveau triés par ordre croissant dans la table. Lors de l'échange de deux colonnes j et $j+1$, les seules valeurs de M_1 impactées sont M_{1j} et $M_{1(j+1)}$. La valeur de A^2_{2N} peut être incrémentée en conséquence.

En particulier, quatre cas se présentent lors de l'échange des colonnes j et $j+1$ pour l'incrémentation des valeurs de M_{1j} et $M_{1(j+1)}$ et de A^2_{2N} :

- **Cas 1 :** $Z_j \in S_1$ et $Z_{(j+1)} \in S_1$. Ce cas ne ce présente jamais car l'insertion fait toujours intervenir un élément de S_2.

— **Cas 2 :** $Z_j \in S_2$ et $Z_{(j+1)} \in S_2$. Dans ce cas, M_{1j} et $M_{1(j+1)}$ gardent les mêmes valeurs qu'avant l'échange. De même, A^2_{2N} reste identique.

— **Cas 3 :** $Z_j \in S_1$ et $Z_{(j+1)} \in S_2$. Notons M_{1j} et $M_{1(j+1)}$ les valeurs dans la quatrième ligne des colonnes j et $j+1$ avant échange M'_{1j} et $M'_{1(j+1)}$ les valeurs dans la quatrième ligne des colonnes j et $j+1$ après échange. Dans ce cas, on a :

- $M_{1(j+1)} = M_{1j}$
- $M'_{1j} = M_{1j} - 1$
- $M'_{1(j+1)} = M_{1j}$

De plus, les seuls termes impactés dans la somme de l'équation A.2 sont :

$$a = (\frac{M^2_{1j}}{j(N-j)} + \frac{M^2_{1(j+1)}}{(j+1)(N-(j+1))}) - (\frac{M_{1j}}{N-j} + \frac{M_{1(j+1)}}{N-(j+1)}) \tag{A.4}$$

qui devient :

$$b = (\frac{M'^2_{1j}}{j(N-j)} + \frac{M'^2_{1(j+1)}}{(j+1)(N-(j+1))}) - (\frac{M'_{1j}}{N-j} + \frac{M'_{1(j+1)}}{N-(j+1)}) \tag{A.5}$$

$$= (\frac{(M_{1j}-1)^2}{j(N-j)} + \frac{M^2_{1j}}{(j+1)(N-(j+1))}) - (\frac{M_{1j}-1}{N-j} + \frac{M_{1j}}{N-(j+1)}) \tag{A.6}$$

$$= a + \frac{1-2*M_{1j}}{j(N-j)} + \frac{1}{N-j} \tag{A.7}$$

La statistique A^2_{2N} doit donc être incrémentée de $\Delta A = 4(\frac{1-M^2_{1j}}{j(N-j)} + \frac{1}{N-j})$

— **Cas 4 :** $Z_j \in S_2$ et $Z_{(j+1)} \in S_1$. Un raisonnement similaire donne :

- $M_{1(j+1)} = M_{1j} + 1$
- $M'_{1j} = M_{1j} + 1$
- $M'_{1(j+1)} = M_{1j}$

et $\Delta A = 4(\frac{1+2*M_{1j}}{j(N-j)} - \frac{1}{N-j})$

La mise à jour de la table A.1 et l'incrémentation de la statistique sont donc réalisées en place et ne font intervenir que $O(N)$ opérations. En particulier, cela évite de retrier les échantillons à chaque déplacement de la fenêtre glissante.

Annexe B

CoreSelect : une heuristique de sélection de prototypes

L'heuristique CoreSelect proposée ici est une variante de l'heuristique dselect proposée par Kar et Jain, 2011. Une pseudocode de cette heuristique est donné par l'algorithme 3. Telle quelle, cette heuristique présente cependant un inconvénient qui peut être expliqué à travers l'exemple présenté figure B.1. Dans ce cas, les données sont groupés suivant trois clusters tels que les points du cluster central sont plus proches des points des clusters de droite et de gauche que les points du cluster de droite ne le sont de ceux de gauche. La dissimilarité est ici la distance euclidienne. Dans ce cas, si les prototypes préalablement sélectionnés ne sont pas alignés, il existe un point unique, appelé centre médian, qui minimise la somme des distances à tous ces prototypes. De plus, le problème de minimisation est convexe. Cela signifie que des points proches de ce centre ont peu de chance d'être sélectionnés comme prototypes. L'heuristique dselect favorise donc la sélection de points en marge des données plutôt qu'au centre. C'est ce qui est observé dans l'exemple de la figure B.1, ou l'heuristique dselect ne sélectionne que des données dans les clusters de droite et de gauche mais jamais au centre. Un comportement similaire a été observé dans le jeu de données de scans de billons, qui peuvent être groupées en cinq clusters suivant leur longueur.

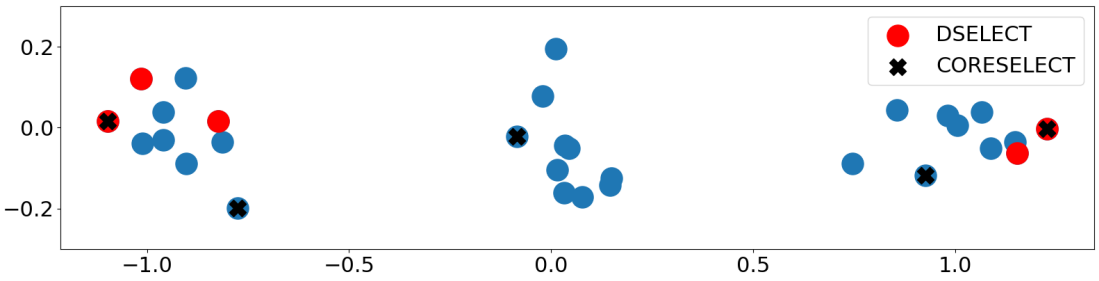

Figure B.1 : Comparaison des prototypes sélectionnés par dselect et coreselect sur un exemple simple

Il est cependant possible de modifier cette heuristique pour combler cette lacune, en sélectionnant

le prochain prototype non pas comme celui maximisant la distance moyenne aux prototypes précédemment sélectionnés, mais celui maximisant la distance minimale aux prototypes précédemment sélectionnés. Un pseudocode de cette heuristique, nommée ici coreselect est présenté algorithme 4. Une motivation à l'utilisation de cette heuristique est que dans le cas ou la dissimilarité est une fonction distance, elle donne une solution approché peu coûteuse à un problème des m centres (Dyer et Frieze, 1985). L'énoncé de ce problème est le suivant. Étant donné un ensemble de N points, chercher m points centraux tels que si des clusters sont définis en associant chacun des N points au point central le plus proches, le rayon maximum des clusters est minimal :

$$\min_{R=(x_1,...,x_m)\subset X} \Delta(x_1, ..., x_m)\,, \tag{B.1}$$

avec

$$\Delta(x_1, ..., x_m) = \max_{x\in X} \min_{R} d(x, r_i)\,. \tag{B.2}$$

Cette heuristique produit 2-optimale à ce problème, au sens que la valeur obtenue pour Δ avec le groupe de prototypes sélectionnés par cette approche est au plus deux fois la valeur optimale.

Algorithm 3 Dselect

Input $X = (x_1, ..., x_n)$, exemples d'entraînement
Output $R = (r_1, ..., r_q)$, ensemble des prototypes

$r_1 \leftarrow$ élément aléatoire choisit dans X
for $i \in [\![1, q]\!]$ **do**
  $r_i \leftarrow argmax_{x\in X\setminus(r_1,...,r_{i-1})}(min_{r_j\in(r_1,...,r_{i-1})}(d(x, r_j))$
end for

Algorithm 4 CoreSelect

Input $X = (x_1, ..., x_n)$, set training inputs
Output $R = (r_1, ..., r_q)$, set of landmarks

$r_1 \leftarrow$ random element from X
for $i \in [\![1, q]\!]$ **do**
  $r_i \leftarrow argmax_{x\in X\setminus(r_1,...,r_{i-1})}(min_{r_j\in(r_1,...,r_{i-1})}(d(x, r_j))$
end for

Une comparaison des erreurs quadratiques moyennes commises par des modèles de substitution de simulateurs de sciage pour des prototypes sélectionnés aléatoirement, avec Coreselect, ou avec dselect est présentée table B.1. Ces modèles sont des forêts aléatoire entraînées sur 500 exemples de la base de données présentée dans le chapitre 5 et sont évaluées sur les 1719 exemples restant. Plusieurs valeurs du paramètre *max_features* sont utilisées. Pour chaque méthode 100 prototypes sont sélectionnés dans la base d'entraînement. Ces résultats sont comparés à ceux obtenus avec les caractéristiques métiers. Les résultats présentés sont moyennés sur 100 répétitions de la séparation du jeu de données en bases d'entraînement et de test.

max features	**CoreSelect**	**Dselect**
1.0	3.16 (0.10)	3.14 (0.10)
0.5	3.13 (0.09)	3.16 (0.11)
0.1	**3.10 (0.08)**	3.13 (0.10)

TABLE B.1 : Moyenne et déviation standard sur 100 répétition des MSE obtenus avec CoreSelect et Dselect.

Tous les modèles de substitution utilisant des caractéristiques calculées à partir de la dissimilarité ICP ont une erreur quadratique moyenne inférieure à celle des modèles de substitution utilisant des caractéristiques métiers. Dans l'ensemble, les erreurs quadratiques moyennes les plus faibles sont obtenues à partir des caractéristiques calculées en tant que dissimilarités par rapport aux repères sélectionnés à l'aide de l'heuristique CORESELECT, pour $max\_features = 0.1$. En particulier, un test de Student appairé comparant les moyennes des RMSE obtenues sur les différentes séparations train-test avec CoreSelect et Dselect pour cette valeur de $max\_features$ donne une p-value de 1.4×10^{-9}. De même, pour $max\_features = 0.5$ et $max\_features = 1.0$, ces p-values sont respectivement 3.3×10^{-5} et 5.2×10^{-4}. Ainsi, au moins sur ce jeu de donnée et avec ce modèle d'apprentissage automatique, CoreSelect permet d'obtenir des MSE plus faibles que Dselect.

Annexe C

Résultats détaillés des expériences numériques présentées dans la section du chapitre

Cette annexe présente les résultats détaillés des expériences réalisées dans la section 3.5.3 sur les méthodes de détection de dérives. En particulier, la table C.1 présente, pour chaque méthode de détection de dérives et jeu de données de benchmark, le taux de détection de dérives obtenu sur les 100 répétitions des expériences. On observe déjà, sur le jeu de donnée WQ qui est le plus petit de tous, un taux de détection plus faible pour la stratégie de couplage échantillonnant activement le flux. Ce taux de détection plus faible peut être expliqué par les temps de détection de dérives plus important mis en évidence dans le chapitre 3. En effet, si le flux de données termine avant que la dérive ne soit terminée, cela augmente le taux de non détection.

Les temps de détection médian par méthode de détection et jeu de données sont présentés dans la table C.2. Les comparaisons de ces temps entre les échantillonnages actifs et passifs sont détaillées dans le chapitre 3. De même, les nombres moyens de fausses détections sont présentés dans la table C.3. Dans ce cas, cependant, les faibles valeurs observées rendent difficile l'interprétation de ces résultats.

			WQ	AE	CCPP	GT	EG	SD
Kolmogorov-Smirnov	$\rho = 0.005$	**AL**	0.80	0.85	1.0	0.99	0.99	1.0
		RD	0.98	0.91	0.99	1.0	1.0	1.0
	$\rho = 0.001$	**AL**	0.55	0.71	1.0	1.0	1.0	1.0
		RD	0.92	0.77	1.0	1.0	1.0	1.0
Anderson-Darling	$\rho = 0.005$	**AL**	0.84	0.77	0.98	1.0	0.99	1.0
		RD	0.96	0.90	0.99	1.0	0.99	0.99
	$\rho = 0.001$	**AL**	0.53	0.68	0.98	0.99	0.99	0.99
		RD	0.92	0.80	0.99	1.0	0.99	1.0
Page-Hinkley	$h = 10\sigma$	**AL**	0.57	0.35	0.94	0.98	0.98	0.98
		RD	0.75	0.49	0.97	0.97	0.99	1.0
	$h = 15\sigma$	**AL**	0.53	0.22	0.98	0.97	1.0	0.99
		RD	0.82	0.37	0.98	0.94	0.96	1.0
Carte de contrôle		**AL**	1.0	1.0	0.99	1.0	1.0	1.0
		RD	0.98	1.0	1.0	1.0	1.0	1.0

TABLE C.1 : Taux de détection de dérive sur 30 répétitions, pour quatre méthodes de détection et six jeux de données UCI

			WQ	AE	CCPP	GT	EG	SD
Kolmogorov-Smirnov	$\rho = 0.005$	**AL**	709 (572)	1278 (1696)	184 (72)	241 (100)	301 (94)	243 (88)
		RD	479 (148)	1321 (2036)	187 (62)	207 (72)	260 (82)	251 (84)
	$\rho = 0.001$	**AL**	726 (596)	1931 (3982)	219 (56)	263 (98)	341 (110)	288 (98)
		RD	615 (530)	1400 (2934)	211 (53)	229 (70)	291 (108)	275 (83)
Anderson-Darling	$\rho = 0.005$	**AL**	772 (923)	2152 (2955)	174 (65)	234 (94)	248 (90)	233 (92)
		RD	442 (253)	1177 (2631)	169 (62)	195 (60)	228 (101)	229 (78)
	$\rho = 0.001$	**AL**	771 (658)	1623 (2733)	190 (67)	245 (88)	307 (120)	241 (85)
		RD	551 (614)	2063 (2742)	187 (62)	206 (77)	252 (78)	256 (83)
Page-Hinkley	$h = 10\sigma$	**AL**	553 (1014)	2256 (3008)	34 (50)	103 (167)	74 (52)	77 (49)
		RD	295 (314)	1733 (4010)	24 (28)	93 (78)	50 (45)	55 (53)
	$h = 15\sigma$	**AL**	445 (883)	2534 (4084)	31 (29)	105 (127)	95 (63)	84 (51)
		RD	238 (447)	2800 (4553)	35 (34)	111 (101)	64 (46)	78 (48)
Carte de contrôle		**AL**	405 (437)	361 (375)	398 (823)	346 (345)	234 (395)	233 (577)
		RD	359 (433)	323 (289)	295 (486)	374 (327)	204 (506)	321 (562)

TABLE C.2 : Temps de détection médian sur 30 répétitions, pour quatre méthodes de détection et six jeux de données UCI. Les écarts inter-quartiles sont également indiqués entre parenthèse.

			WQ	AE	CCPP	GT	EG	SD
Kolmogorov-Smirnov	$\rho = 0.005$	**AL**	0.11 (0.31)	0.98 (0.92)	0.32 (0.60)	0.22 (0.46)	0.21 (0.45)	0.74 (0.75)
		RD	0.15 (0.38)	0.53 (0.76)	0.32 (0.53)	0.18 (0.38)	0.31 (0.58)	0.66 (0.78)
	$\rho = 0.001$	**AL**	0.04 (0.20)	0.39 (0.63)	0.10 (0.33)	0.10 (0.33)	0.07 (0.26)	0.35 (0.61)
		RD	0.04 (0.20)	0.11 (0.31)	0.07 (0.29)	0.04 (0.20)	0.08 (0.27)	0.12 (0.35)
Anderson-Darling	$\rho = 0.005$	**AL**	0.14 (0.35)	0.69 (0.81)	0.27 (0.52)	0.23 (0.47)	0.22 (0.46)	0.63 (0.78)
		RD	0.13 (0.36)	0.32 (0.63)	0.29 (0.52)	0.13 (0.36)	0.49 (0.77)	0.53 (0.75)
	$\rho = 0.001$	**AL**	0.03 (0.17)	0.24 (0.45)	0.07 (0.26)	0.07 (0.26)	0.11 (0.31)	0.20 (0.40)
		RD	0.01 (0.10)	0.14 (0.34)	0.04 (0.20)	0.04 (0.20)	0.08 (0.27)	0.16 (0.39)
Page-Hinkley	$h = 10\sigma$	**AL**	1.20 (1.03)	4.68 (4.16)	2.42 (1.72)	2.92 (1.78)	1.63 (1.46)	2.61 (1.85)
		RD	1.32 (1.13)	5.45 (4.18)	2.49 (1.53)	3.74 (2.09)	1.50 (1.22)	4.10 (2.82)
	$h = 15\sigma$	**AL**	0.43 (0.65)	2.19 (2.47)	1.20 (1.05)	1.38 (1.18)	0.54 (0.73)	0.93 (0.95)
		RD	0.60 (0.72)	2.83 (2.90)	1.75 (1.30)	2.33 (1.56)	0.46 (0.62)	1.75 (1.35)
Carte de contrôle		**AL**	3.78 (0.87)	28.55 (1.62)	7.13 (1.50)	8.65 (0.92)	7.75 (1.49)	14.71 (2.80)
		RD	4.04 (0.88)	31.13 (0.88)	7.55 (142)	9.1 (0.8)	7.73 (1.54)	21.84 (2.43)

TABLE C.3 : Nombre moyen de fausses alertes par flux, sur 30 répétitions, pour quatre méthodes de détection et six jeux de données UCI. Les écarts-types sont également indiqués entre parenthèse.

Abstract

Toward industrial digital shadows and twins : a novel strategy to couple simulation and machine learning models, applied to the lumber industry

This thesis is part of the ANR project Lorraine-Artificial Intelligence, a multi-disciplinary project promoting research into both artificial intelligence itself, and its applications to other fields. As such, this thesis focuses on the development and use of machine learning models as a substitute for simulation models. Interest in this research topic is fueled by academic and industrial interest in the concept of digital shadows and twins, seen as an evolution of simulation models for long-term use at the heart of systems and processes.

The main contribution of this thesis is the proposal of a coupling strategy between a simulation model and a surrogate model performing the same prediction task repeatedly on a data stream. The simulation model is assumed to have a high level of fidelity, but to be too slow or computationally expensive to be used alone to perform the full range of prediction required. The surrogate model is a fast machine-learning model that approximates the simulation model. The primary objective of the proposed coupling strategy is the efficient use of limited computational resources by intelligently allocating each prediction request to one of the two models. This allocation is, in particular, inspired by active learning and based on the evaluation of the level of confidence in the predictions of the machine learning model. Numerical experiments are first carried out on eight datasets from the scientific literature. An application to the sawmilling industry is then developed.

Keywords *Sawmill*, *Machine learning*, *active learning*,*Digital Shadow*, *Surrogate modelling.*

Résumé

Contributions aux ombres et jumeaux numériques dans l'industrie : proposition d'une stratégie de couplage entre modèles de simulation et d'apprentissage automatique appliquée aux scieries

Ces travaux de thèse s'inscrivent dans le projet ANR Lorraine-Intelligence Artificielle qui se veut un projet multi-disciplinaire promouvant la recherche à la fois sur l'intelligence artificielle elle-même et sur ses applications à d'autres domaines de recherche. A ce titre, cette thèse s'intéresse au développement et à l'utilisation de modèles d'apprentissage automatique comme modèles de substitution à des modèles de simulation. L'intérêt pour ce sujet de recherche est, en particulier, porté par l'engouement des milieux académiques et industriels pour le concept d'ombres et jumeaux numériques, vus comme une évolution des modèles de simulation pour une utilisation pérenne au cœur des systèmes et des processus.

La contribution principale de ces travaux de thèse est la proposition d'une stratégie de couplage entre un modèle de simulation et un modèle de substitution réalisant une même tâche de prédiction de manière répétée sur un flux de données. Le modèle de simulation est supposé avoir un haut niveau de fidélité mais être trop lent ou coûteux en calcul pour être utilisé seul pour réaliser l'intégralité des prédictions requises. Le modèle de substitution est un modèle d'apprentissage automatique qui approxime le modèle de simulation. L'objectif premier de la stratégie de couplage proposée est l'utilisation efficiente des ressources en calcul limitées par l'allocation intelligente de chaque prédiction à effectuer à un des deux modèles. Cette allocation est, en particulier, inspirée de l'apprentissage actif et basée sur l'évaluation de niveaux de confiance dans les prédictions du modèle d'apprentissage automatique. Des expériences numériques sont d'abord menées sur huit jeux de données de la littérature scientifique. Une application à l'industrie du sciage est ensuite développée.

Mots-clés*Scierie*, *Apprentissage automatique*, *Apprentissage actif*, *Ombre numériques*, *Modèles de substitution.*

UNIVERSITÉ DE LORRAINE

cnrs

CRAN