Détection d'anomalies dans les flux de données pour une application dans les réseaux de capteurs

Kévin Ducharlet

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Institut National des Sciences Appliquées de Toulouse

Présentée et soutenue par

Kévin DUCHARLET

Le 28 septembre 2023

Détection d'anomalies dans les flux de données pour une application dans les réseaux de capteurs

Ecole doctorale : **SYSTEMES**

Spécialité : **Informatique et Automatique**

Unité de recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Louise TRAVE-MASSUYES et Marie-Véronique LE LANN

Jury

M. Christophe BERENGUER, Rapporteur
Mme Anne SABOURIN, Rapporteure
Mme Louise TRAVE-MASSUYES, Directrice de thèse
Mme Marie-Véronique LE LANN, Co-directrice de thèse
M. Vincent COCQUEMPOT, Président

Remerciements

Avant de débuter ce manuscrit, je tiens à remercier les personnes qui m'ont accompagné et soutenu au cours de cette thèse. J'aurais aimé faire davantage de rencontres, mais le contexte sanitaire n'était pas favorable aux conférences et autres colloques de chercheurs. Cependant, je me réjouis de la qualité des rencontres que j'ai pu faire.

D'abord, je souhaite remercier mes encadrants ; mes deux directrices de thèse, Louise Travé-Massuyès et Marie-Véronique Le Lann, ainsi que mon encadrant industriel, Youssef Miloudi. Ils ont tous les trois su être présents et de bons conseils tout en me laissant les libertés nécessaires pour que je puisse aujourd'hui considérer ces travaux comme les miens.

Je remercie aussi grandement Jean-Bernard Lasserre qui m'a accompagné dans la compréhension de la fonction de Christoffel et sans qui une partie des contributions de cette thèse n'aurait pas été possible.

Il me semble également important de remercier mes rapporteurs, Anne Sabourin et Christophe Berenguer, qui ont pris le temps de relire ce manuscrit rigoureusement et avec intérêt, soulevant des défauts qu'un manque de recul m'empêchait jusqu'alors de percevoir. Ce manuscrit est aujourd'hui plus complet grâce à eux. Je remercie aussi mon examinateur et président du jury de soutenance, Vincent Cocquempot, pour avoir également accepté d'évaluer mes travaux.

Au cours de ma thèse, j'ai eu la chance d'être en contact avec différents docteurs qui m'ont apporté leur soutien et un retour d'expérience très utile d'un point de vue méthodologique, mais aussi moral. Je remercie ainsi Liwen Zhang, Stephanie Rey, Christophe Bortolaso, Katarzyna Borgiel, Florent Mouysset, Mehdi Kandi, Sara Maqrot et Mokhtar Boumedyen Billami.

J'ai également côtoyé de nombreux doctorants avec qui j'ai partagé l'expérience de la thèse, des expériences positives comme des expériences négatives, mais surtout beaucoup d'expériences humaines. Merci à l'équipe des "bébous", Marion Olivier, Julien Breton, Nihed Bendahman et Camille Gosset, et désolé pour les incitations à la procrastination. Merci à Hamza Safri, Ikram Boukharouba, Elodie Toufaili, et également à Adrien Dorise, Le Toan Duong, Charlotte Lacoquelle, Louis Goupil, Edgar Sepulveda Oviedo et Camille Coquand. Et merci enfin aux plus éloignés Baptiste Chopin et Nicolas Ranger.

Je tiens aussi à remercier mes autres collègues comme ma drôle d'amie Lina Nicolaieff, Sebastien Dufour, qui aura malmené mes contributions, et mes camarades de la DIAC, Ugo Mekki et Florian Janela.

Je remercie tous mes amis là-haut dans le Nord. Vous m'avez plus souvent fait douter que vous ne m'avez supporté, mais je ne vous en veux pas. Merci à Vivien Bernard de ne pas m'avoir entraîné dans l'abandon, merci à mes comparses de la Trinité, Rémi Tanfin et Thomas Sueur, et aux autres membres de la Team M, Martin Farissier et Thomas Joly, merci enfin à Romain De Graaf, Gabrielle Eteve, Alexis Bridoux et Arthur Duvinage.

Je remercie bien sûr ma famille, en commençant par mes parents qui ont toujours

été là pour me soutenir, et je rends en particulier hommage à mon père que j'ai perdu au cours de cette thèse, j'aurais aimé que tu sois là pour en célébrer la fin. Je remercie également ma grande soeur, ainsi que ma grand-mère et ma grande-tante pour tous leurs encouragements.

Enfin, cette thèse n'aurait pas vu le jour sans Mustapha Derras, que je remercie à la fois d'avoir permis le financement de mes travaux, mais aussi d'avoir donné à l'entreprise Berger-Levrault l'image d'un industriel voyant la recherche autrement que la justification du CIR.

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

CPS Système cyber-physique (Cyber-Physical System)

DB Base de données (database)

DL Apprentissage profond (Deep Learning)

DyCF Dynamical Christoffel Function

DyCG Dynamical Christoffel Growth

FCE Fonction de Christoffel empirique

FC Fonction de Christoffel

F-mesure $F_{measure} = 2 \times \frac{R \times P}{R+P}$

FN Faux négatifs

FP Faux positifs

Taux de faux positifs $FPR = \frac{FP}{VN+FP}$

GMAO Gestion de Maintenance Assistée par Ordinateur

GMM Modèle de mélange gaussien (Gaussian Mixture Model)

IIoT IoT pour l'industrie

IoT Internet des Objets (Internet of Things)

KDE Estimation de densité par noyau (Kernel Density Estimation)

kNN k plus proches voisins

k^{th}NN k-ième plus proche voisin

LOF Local Outlier Factor

ML Apprentissage automatique (Machine Learning)

NCD Noyau de Christoffel-Darboux

pdf Fonction densité de probabilité (Probability density function)

Précision $P = \frac{VP}{VP+FP}$

Rappel $R = \frac{VP}{VP+FN}$

SuMeLInd SuMeLI avec application indépendante des méthodes

SuMeLInk SuMeLI avec lien entre les méthodes appliquées successivement

SuMeLI Successive Methods Learned Incrementally

SuMeRI Successive Methods Run Iteratively

VN Vrais négatifs

VP Vrais positifs

WOLF-DyCF Unification de WOLF avec DyCF

WOLF-Eval Méthode d'évaluation pour WOLF

WOLF-KDE Unification de WOLF avec les KDE

WOLF WSNs OutLier detection Framework

WSN Réseau de capteur (Wireless Sensor Network)

Chapitre 1

Introduction

Les travaux réalisés dans cette thèse ont pour objectif premier de *réaliser une analyse de la fiabilité des données issues de réseaux de capteurs dans un vaste panel de cas industriels et en amont d'une phase de diagnostic plus poussée.* Du fait de la diversité des cas d'application, il serait coûteux d'adopter une approche reposant sur des connaissances métier, et il lui est donc préférée une approche basée sur les données ; pour analyser leur fiabilité, cette approche sera celle de la détection d'anomalies.
Ce premier chapitre présente successivement les différents concepts nécessaires pour la compréhension des problématiques et approches abordées au sein de cette thèse, introduites dans la Section 1.5.

Sommaire

1.1 Présentation des réseaux de capteurs

Présentons tout d'abord le concept des *réseaux de capteurs*, qui ne peut être dissocié du contexte de l'*Internet des Objets*, à travers leurs enjeux et leurs spécificités.

1.1.1 Introduction à l'Internet des Objets pour l'industrie

1.1.1.1 Histoire

La première utilisation du terme Internet des Objets (IoT) est attribuée au britannique Kevin Ashton en 1999 pour décrire un système dans lequel des objets physiques sont connectés à Internet via l'usage de capteurs. L'IoT s'est par la suite popularisé pour décrire une situation dans laquelle un capteur, un appareil ou un simple objet du quotidien est équipé d'une capacité de calcul et de connexion à Internet [Rose 2015].

Son développement est principalement dû à celui de différentes technologies permettant :

— la connectivité à faibles coûts de n'importe quel type d'objet,
— la standardisation de l'adressage IP dans les réseaux,
— le développement de capacités de calcul à des coûts de plus en plus faibles,
— la miniaturisation des appareils permettant l'incorporation des technologies précédentes dans des objets de plus en plus petits,
— les avancées en analyse de données,
— et enfin l'émergence du calcul distribué dans les réseaux.

Ainsi, l'IoT a rapidement pu se propager dans de nombreux cas d'application comme la santé, notamment avec des appareils portables telles que les montres connectées ou des appareils médicaux pouvant être ingérés par les patients et permettant le suivi de leur état de santé et de bien-être, la gestion des villes et des bâtiments, avec un suivi de la consommation énergétique, une gestion de la température ou encore le contrôle des éclairages, l'industrie, avec la maintenance des équipements, le suivi des cycles de vie et de fabrication, ou encore dans les véhicules, l'agriculture, etc.

1.1.1.2 Industrie 4.0

Tandis que les trois premières révolutions industrielles portent principalement sur la production mécanique – d'abord avec l'utilisation de la puissance hydraulique et les machines à vapeur, puis avec l'électricité, la production en masse et les lignes d'assemblage, et enfin grâce à l'automatisation et l'informatique – la quatrième révolution industrielle, apparaissant en 2011 sous le terme de Industrie 4.0, se concentre sur l'organisation plus intelligente des moyens de production.

Plus exactement, l'Industrie 4.0 s'appuie sur l'utilisation de systèmes cyber-physiques (CPS). Il s'agit de systèmes combinant des composants physiques et des composants virtuels et ayant des capacités de calcul, de mesure, de contrôle et de

communication. Dans l'Industrie 4.0, ces CPS permettent la prise autonome de décisions et l'amélioration de l'efficacité, de la productivité, de la sécurité et de la transparence dans l'industrie [Boyes 2018].

1.1.1.3 Internet des Objets pour l'industrie

La description de l'Industrie 4.0 précédemment nommée mène naturellement à l'IoT industrielle (IIoT), définie par [Boyes 2018] comme "un système comprenant des objets connectés, des équipements cyber-physiques, les technologies d'informations génériques associées et, éventuellement, des plates-formes de calcul en ligne ou distribuées. Ce système permet l'accès, la collecte, l'analyse, la communication et l'échange, en temps réel et de manière intelligente et autonome, d'un processus, d'un produit ou d'un service d'information, ceci afin d'optimiser la production de valeur. Cette valeur peut être comprise comme l'amélioration d'un produit ou d'un service délivré, un gain en productivité, la réduction du cycle de production, du coût des tâches ou de la consommation énergétique".

Ainsi, le développement de l'IIoT a conduit au déploiement quasiment systématique de capteurs, connectés au sein d'un réseau sans fil, sur des systèmes industriels, avec des objectifs d'amélioration des performances et de réduction des coûts.

1.1.2 Concept des réseaux de capteurs

1.1.2.1 Définition

A travers la littérature, plusieurs définitions sont proposées pour les réseaux de capteurs (WSNs) :

— "un grand nombre de noeuds capteurs, densément déployés à l'intérieur d'un phénomène ou proches de celui-ci. La position de ces noeuds n'a pas besoin d'être pré-conçue, (...) ce qui signifie que les protocoles et algorithmes doivent posséder des capacités d'auto-organisation. Une autre caractéristique unique des WSNs est la capacité des noeuds à collaborer. Les noeuds sont également équipés d'un processeur, leur permettant de réaliser des calculs simples et de ne transmettre que les données nécessaires." [Akyildiz 2002]

— "un ensemble de noeuds capteurs interconnectés qui sont capables de surveiller l'environnement, fournir des mesures et lever des alertes selon des indicateurs afin de réaliser l'analyse de l'environnement. Les cas d'applications sont extensibles et le nombre de noeuds peut varier d'une centaine à des milliers pendant le temps de fonctionnement." [Jemal 2013]

— "généralement composés de noeuds capteurs à faible puissance, faible coût et restreints en énergie, qui sont déployés pour mesurer un phénomène physique d'intérêt. Les données sont collectées par des points d'accès. A partir des données collectées, une application est conçue pour surveiller et/ou contrôler le monde physique. (...) Les noeuds embarquent des capacités de calcul et des capacités mémoires, bien que possiblement limitées." [Mokrenko 2014]

— “une collection de noeuds sans fils avec des capteurs multi-fonctions collaborant pour surveiller une zone assignée en accomplissant des tâches de mesures dans un environnement changeant dynamiquement. La surveillance et le suivi sont ainsi deux champs d’application majeurs pour les WSNs.” [Liu 2016]

De ces différentes définitions, retenons que les WSNs *se composent d’un ensemble de noeuds capteurs et que chaque noeud capteur a la capacité de mesurer l’environnement, embarque une capacité de calcul et une capacité mémoire lui permettant de réaliser des tâches de pré-calcul simples sur les mesures relevées et peut transmettre les données dans un réseau qui permet la collaboration de différents noeuds capteurs.*

1.1.2.2 Enjeux industriels

Les enjeux des WSNs dans l’industrie sont généralement similaires à ceux découlant de la définition de l’IIoT.

Dans le cadre industriel de cette thèse, les WSNs sont notamment utilisés pour la gestion de maintenance assistée par ordinateur (GMAO), dans un contexte de jumeaux numériques des équipements, mais également pour la gestion des bâtiments. Dans ces deux cas, l’objectif principal est de permettre la réduction de coûts, que ce soit à travers une maintenance prévisionnelle sur des systèmes industriels, moins coûteuse en interventions qu’une maintenance préventive et évitant les lourds coûts de la maintenance corrective, ou à travers l’analyse du comportement des utilisateurs d’un bâtiment pour réduire la consommation énergétique.

Les applications mentionnées nécessitent de relever des mesures précises sur les systèmes d’intérêt pour en étudier le comportement, ce que permettent de réaliser les WSNs à faible coût et avec une fiabilité relative motivant les travaux de cette thèse.

1.1.2.3 Spécificités des noeuds capteurs

Les définitions présentées pour les WSNs mettent toutes en avant les équipements qui composent ces réseaux, à savoir les *noeuds capteurs*. Ces objets possèdent différentes spécificités dont il faut tenir compte dans leur étude :

— une capacité de calcul limitée ; bien que les WSNs soient capables de réaliser des calculs en périphérie grâce aux processeurs embarqués par les noeuds capteurs, ces calculs sont limités par la faible puissance des processeurs, ce qui est dû au faible coût des capteurs et justifié par leur grand nombre,

— une capacité mémoire limitée ; les noeuds capteurs ne peuvent pas non plus stocker les mesures relevées sur de longues périodes, ni des modèles d’apprentissage trop lourds, forçant l’utilisation de modèles légers ou à laisser à des noeuds plus performants le soin de réaliser les tâches nécessitant le stockage d’un grand nombre de mesures ou de ces modèles lourds,

— une source d’énergie limitée ; les noeuds capteurs sont souvent alimentés par des batteries dont la durée de vie est grandement dépendante de l’usage fait

de ces capteurs. Ainsi, un noeud capteur dont les capacités de calcul sont souvent sollicitées, ou qui transmet un grand nombre de mesures dans le réseau, verra la durée de vie de sa batterie diminuer bien plus rapidement, pouvant causer des dysfonctionnements du capteur.

Finalement, ces spécificités entraînent un compromis pour la sauvegarde de la batterie entre (1) la transmission des mesures dans le réseau pour une analyse plus centralisée et (2) des calculs, ne pouvant être trop lourds, réalisés en périphérie afin de limiter les communications. Notons également que le coût de communication par bit peut être dans certains cas 10^3 à 10^4 fois plus élevé que celui de calcul [Zhao 2003] et que des méthodes pour augmenter la capacité de stockage des noeuds capteurs existent [Mathur 2006].

1.1.2.4 Modèle d'architecture

Cette étude ne portant pas sur l'aspect technique des WSNs, il n'en sera présenté ici qu'une architecture très générique offrant un support visuel à cette section. La Figure 1.1 présente cette architecture, dans laquelle des ensembles de noeuds capteurs sont regroupés en champs de capteurs.

Les noeuds au sein de chaque champ peuvent communiquer entre eux et les données sont finalement transmises à un noeud récepteur, possédant des capacités plus importantes que les noeuds capteurs.

Les noeuds récepteurs transmettent les données à un serveur pour les stocker dans une base de données (DB) à travers internet en passant par une station de base. C'est au niveau des serveurs que les calculs les plus lourds peuvent être réalisés, et l'utilisateur final accède aux données à travers une interface connectée à la DB.

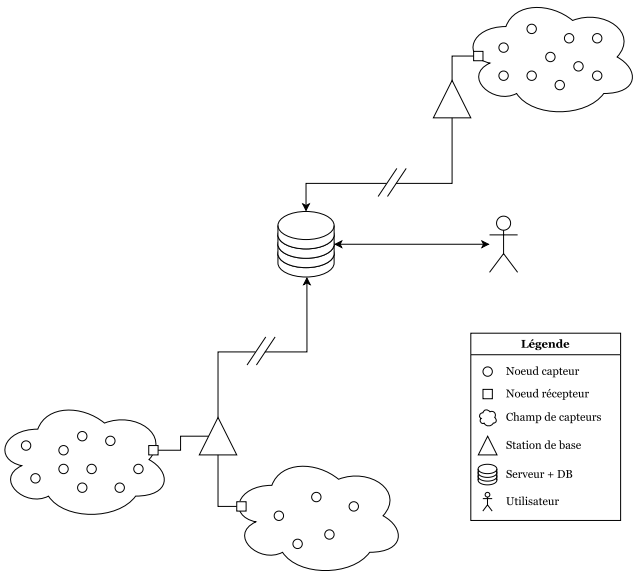

Figure 1.1 – Représentation générique d'un réseau de capteur (WSN)

1.2 Description du problème de détection d'anomalies

Cette section présente le domaine de la détection d'anomalies, retenu dans l'étude des WSNs. Sans en fournir un état de l'art, elle introduit un grand nombre de concepts importants pour la compréhension des travaux réalisés dans cette thèse. La Section 1.2.3 précise quant à elle l'intérêt de la détection d'anomalies dans les WSNs.

1.2.1 Qu'est-ce que la détection d'anomalies ?

1.2.1.1 Histoire et définition

La détection d'anomalies, qui appartient au plus large domaine de la fouille de données, est un sujet qui intéresse diverses communautés, à commencer par celle des statisticiens, depuis plusieurs siècles. Il est généralement admis que les travaux de Edgeworth [F. Y. Edgeworth 1887], à la fin du XIXe siècle, sur l'étude d'observations discordantes, fournissaient une première définition formelle de l'anomalie et donnaient naissance au domaine. Cependant, la présence de telles observations étant commune dans un grand nombre de domaines plus anciens, il ne serait pas surprenant que leur étude ait commencé bien avant.

Depuis, de nombreux qualificatifs ont été employés pour désigner les observations anormales ; valeurs aberrantes ou rares, observations atypiques ou inattendues, mesures erronées ou frauduleuses, échantillon en dehors de la distribution, sont autant de synonymes au terme "anomalie".

Donner une définition de l'anomalie n'est pas une tâche aisée tant elle est dépendante du contexte d'étude. A travers l'état de l'art, plusieurs définitions ont été formulées, mais les plus communes, et faisant autorité dans le domaine, sont les suivantes :

— [Hawkins 1980] : "une observation qui dévie des autres observations au point de pouvoir être suspectée d'avoir été générée par un mécanisme différent",
— [Barnett 1994] : "une observation, ou un sous-ensemble d'observations, paraissant incohérentes par rapport au reste du jeu de données",
— [Chandola 2009] : "motifs dans les données qui ne sont pas conformes au comportement attendu".

Ainsi, la détection d'anomalies est le domaine consistant à détecter ces anomalies. Il s'apparente à d'autres domaines tels que la détection de nouveauté, visant à détecter de nouveaux comportements potentiellement normaux, ou à la détection de rupture, visant à détecter un changement, souvent brutal, dans la distribution.

Il est important de noter que les définitions fournies sont générales et subjectives, employant des expressions telles que "pouvoir être suspectée de", "paraissant incohérente" et "comportement attendu". Alors que de nouvelles approches ne cessent de voir le jour, cherchant à être meilleures que l'état de l'art en termes de précision de la détection, [Zimek 2018] nous rappelle très justement que de telles approches ne font que mettre en évidence des observations suspectes, et ce, selon une conception spécifique de la normalité. La définition de l'anomalie retenue dans cette thèse

sera celle fournie dans la récente étude de [Ruff 2021] :

Définition 1 *Une anomalie est une observation déviant considérablement d'un certain concept de normalité.*

1.2.1.2 Applications

Les anomalies trouvant leur origine dans les erreurs humaines, naturelles ou frauduleuses, dans les erreurs des instruments ou dans les changements de comportements, la détection d'anomalies trouve une grande variété de domaines d'applications. En s'appuyant sur celles fournies par [Hodge 2004] et [Chandola 2009], une liste non exhaustive des principales applications peut être :

- la détection de fraudes, dans les secteurs des banques, des assurances ou encore des téléphones mobiles, à la recherche d'activités frauduleuses et/ou d'usurpation d'identité,
- la détection d'intrusion, dans le domaine de la cybersécurité, cherchant à détecter des accès non autorisés à des périphériques au sein d'un réseau,
- le diagnostic de fautes ou de défauts, dans le secteur de l'industrie et en lien avec le domaine de recherche du diagnostic, qui suit un processus industriel avec pour objectif de déterminer une erreur de fonctionnement d'un équipement ou des défauts dans la production,
- l'analyse d'images, dans des contextes de vidéo-surveillance, d'imagerie satellite ou d'imagerie médicale, visant à détecter des mouvements inhabituels ou la présence d'éléments anormaux,
- la surveillance médicale pour suivre l'état de santé des patients, avec comme exemple le plus simple le suivi du rythme cardiaque,
- la détection de nouveautés dans des données textuelles, ayant pour objectif d'isoler de nouveaux sujets en tendance par exemple,
- la détection d'anomalies dans les WSNs, développée dans la Section 1.2.3.

Chacun de ces cas d'application amène son lot de spécificités, ainsi le domaine de la détection d'anomalies se décline en un grand nombre de types d'approches, à tel point qu'il peut être difficile à cerner. Aussi, diverses études ont vu le jour en présentant la détection d'anomalies du point de vue de cas d'applications spécifiques [Martin 2007, Auslander 2011].

1.2.2 Taxonomie de la détection d'anomalies

1.2.2.1 Formulation du problème

En présentant les différents aspects de la détection d'anomalies, [Chandola 2009] fournit une approche pour positionner tout problème de détection d'anomalies par rapport à l'état de l'art.

Nature des données Dans un problème de détection d'anomalies, les données sont représentées par un ensemble d'éléments appelés *instances*, et l'objectif est de

déterminer les instances anormales au sein d'un sous-ensemble pour lequel on ne possède pas d'informations sur le caractère anormal des instances (ce sous-ensemble peut être l'ensemble complet).

Chaque instance possède un même nombre d'attributs, aussi appelés caractéristiques ou variables. Dans le cas d'un seul attribut, on parle de problème univarié, et dans le cas de plusieurs attributs, on parle de problème multivarié. Enfin, dans le cas d'un grand nombre d'attributs, on parlera d'un problème en grandes dimensions (traitement d'images ou de données textuelles), et d'un problème en faibles dimensions lorsque ce nombre est relativement faible, généralement moins de dix attributs. A noter que cette différence marque souvent la frontière entre des approches d'apprentissage profond (DL comme Deep Learning) et des approches statistiques et d'apprentissage plus classiques (ML comme Machine Learning).

Les attributs de chaque instance peuvent être quantitatifs (discrets ou continus) ou qualitatifs (nominaux, ordinaux ou binaires). Les instances peuvent avoir des attributs du même type ou des attributs mixtes, ce qui aura un impact sur les méthodes applicables ou le type de distances à utiliser au sein de certaines méthodes.

Pour finir, les relations entre les instances et les variables sont importantes dans la définition d'un problème de détection d'anomalies. Les relations entre les instances sont souvent liées au type de jeu de données : dans un jeu de données séquentielles, comme une série temporelle ou un flux de données, les instances sont généralement liées temporellement, dans un jeu de données spatiales, les instances proches dans l'espace sont généralement liées, et dans un jeu de données sous forme de graphe, ce sont les noeuds connectés qui possèdent des relations entre eux. Enfin, si les attributs sont corrélés, une étude multivariée est nécessaire, alors que s'ils ne le sont pas, on peut se contenter d'étudier les variables indépendamment.

Types d'anomalies [Chandola 2009] propose trois premiers types d'anomalies, et [Ruff 2021] les complètent avec deux autres catégories issues du développement du DL pour traiter des jeux de données plus complexes :

- *anomalies ponctuelles* : il s'agit du type d'anomalies le plus courant, une instance du jeu de données est anormale par rapport à un concept de normalité déduit du reste du jeu de données
- *anomalies contextuelles*, ou parfois *conditionnelles* : ici une instance est anormale dans un contexte spécifique, lié aux relations entre les instances et les variables mentionnées dans le paragraphe précédent ; par exemple, le contexte peut être temporel, et l'instance est considérée comme anormale par rapport à un concept de normalité déduit des instances ayant un contexte temporel similaire
- *anomalies collectives* : il s'agit d'un groupe d'instances qui sont anormales lorsque considérées collectivement (mais souvent normales si on les considère individuellement) ; ce type d'anomalies apparaît principalement dans le cas de données séquentielles ou une sous-séquence est anormale
- *anomalies sensorielles de bas niveau* : dans le cas d'une organisation hié-

rarchique des données, par exemple en traitement d'images ou de données textuelles, il s'agit des anomalies touchant les niveaux hiérarchiques bas, par exemple des pixels dans une image, des mots dans une phrase ou des capteurs dans un WSN

— *anomalies sémantiques de haut niveau* : dans le même contexte que les anomalies sensorielles de bas niveau, on s'intéresse cette fois aux anomalies touchant les niveaux hiérarchiques hauts, par exemple un type d'objet dans une image, le sens d'une phrase ou le comportement d'un système étudié par des capteurs dans un WSN.

Ces types d'anomalies permettent d'approcher un problème de détection d'anomalies, mais des catégorisations différentes sont souvent utilisées lorsqu'on ajoute des connaissances métier et celles-ci sont plus pertinentes pour l'utilisateur. Aussi, il est dans tous les cas possible de se ramener à une détection d'anomalies ponctuelles à travers une phase de pré-traitement ; une anomalie contextuelle est une anomalie ponctuelle dans un sous-ensemble d'instances modélisant le contexte, une anomalie collective est une anomalie ponctuelle dans un jeu de données constitué de séquences d'instances, etc.

Labellisation Ce que l'on appelle un *label* en détection d'anomalies correspond à l'information de l'état de normalité d'une instance. Généralement, un label a donc deux valeurs possibles, *normal* ou *anomalie*, mais il peut arriver que le label prenne d'autres valeurs dans le cas où on aurait plusieurs catégories de points normaux ou d'anomalies.

La labellisation d'un jeu de données, c'est-à-dire l'action de fournir un label pour chaque instance du jeu de données, est une tâche réalisée par un expert du domaine qui s'avère généralement coûteuse en effort. De plus, il est généralement plus facile d'obtenir un ensemble d'instances considérées comme normales qu'un ensemble d'instances labellisées couvrant tous les comportements anormaux possibles.

Selon la disponibilité des labels, on distingue trois types d'approches d'apprentissage pour la détection d'anomalies :

— l'apprentissage supervisé, applicable dans le cas où un sous-ensemble d'instances est labellisé, contenant des instances normales et des instances anormales : cette approche consiste à générer, suite à une phase d'apprentissage utilisant le sous-ensemble labellisé, un modèle de prédiction pour la ou les classes normales et anormales ; le modèle est ensuite utilisé pour prédire l'état de normalité des instances non labellisées,

— l'apprentissage semi-supervisé, applicable dans le cas où un sous-ensemble d'instances possède un label d'un type uniquement, généralement la classe normale et, dans de rares cas, la classe anormale : l'objectif est alors de générer un modèle généralisant la classe connue ; pour déterminer si un point sans label est normal ou anormal, on regarde alors s'il appartient au modèle généré,

— l'apprentissage non supervisé, applicable dans le cas où aucun label n'est

disponible : l'hypothèse qui est généralement faite dans ce cas de figure est que les anomalies sont bien plus rares et éparses que les instances normales, et l'approche devient alors similaire à une approche semi-supervisée qui ne contiendrait que des instances labellisées comme normales.

L'approche supervisée est théoriquement la plus précise, mais la dépendance à la disponibilité de labels la rend rarement utilisable. En pratique, dans la majorité des cas, c'est l'approche non supervisée qui est retenue, bien que sa précision soit faible et difficilement estimable.

Résultat attendu Il y a généralement deux formes de résultats attendus en sortie d'une solution de détection d'anomalies :

- un *score d'anomalie* ou un *score de normalité* qui correspond au degré d'anormalité ou de normalité d'une instance ; cette information permet notamment de fournir un classement des instances selon leur normalité, et l'utilisateur peut n'étudier que les n instances les plus anormales, aussi appelées *top-n anomalies*,
- un *label*, comme décrit dans le paragraphe précédent, ce qui revient soit à fixer le paramètre n dans le choix des top-n anomalies ou un seuil sur le score d'anomalies, soit à utiliser une méthode dont la sortie est binaire (normal ou anomalie).

1.2.2.2 Catégories d'approches

L'état de l'art de la détection d'anomalies fournit un grand nombre de méthodes, mais également d'études visant à les classifier, ce qui n'est pas une tâche aisée étant donnée leur variété. Le Tableau 1.1 décrit les classifications rencontrées dans certaines de ces études [Chandola 2009, Zhang 2013a, Wang 2019, Ruff 2021], et les paragraphes suivants reprennent ces classifications en expliquant brièvement l'approche de chaque catégorie, sans prétendre à l'exhaustivité.

Méthodes statistiques Les méthodes statistiques reposent sur l'hypothèse que le processus générant les instances normales peut être modélisé par une distribution statistique. Les anomalies sont alors définies comme les instances dont la probabilité d'avoir été générées par la distribution statistique est faible. Dans le cas supervisé, on peut également associer les anomalies à une distribution différente, et comparer la vraisemblance qu'une instance soit issue de la distribution normale plutôt que de la distribution anormale.

Ces méthodes sont généralement divisées en deux catégories :

- les approches *paramétriques*, qui supposent une distribution spécifique à l'avance et recherchent les paramètres de la distribution à partir des données lors d'une phase d'apprentissage, c'est notamment le cas des *modèles de mélange gaussien (GMM)*,

Source	Catégorie	Sous-catégorie
[Chandola 2009]	Classification	Réseaux de neurones
		Réseaux Bayésiens
		Machines à Vecteurs de Support
		Basées règles
	Clustering	-
	Plus proches voisins	Distance au k^{th}NN
		Densité relative
	Statistiques	Paramétriques
		Non paramétriques
	Théorie de l'information	-
	Spectrales	-
[Zhang 2013a]	Statistiques	Paramétriques
		Non paramétriques
	Distance	Voisinnage local
		Distance aux kNN
	Densité	-
	Clustering	Partitionnement
		Hiérarchique
		Basé Densité
		Basé Grilles
	Méthodes en grandes dimensions	-
[Wang 2019]	Densité	-
	Statistiques	Paramétriques
		Non paramétriques
		Autres
	Distance	kNN
		Elagage
		Pour flux de données
	Clustering	Partitionnement
		Hiérarchique
		Basé Densité
		Basé Grilles
		En grandes dimensions
	Ensembliste	-
	Apprentissage	Apprentissage Actif
		En Sous-espace
		Basé Graphes
		DL
[Ruff 2021]	Estimation de densité et Modèles probabilistes	-
	Classification à une classe	-
	Modèles de reconstruction	-

TABLE 1.1 – Présentation des classifications des méthodes de détection d'anomalies pouvant être rencontrées dans certaines études de l'état de l'art

— les approches *non paramétriques*, qui ne supposent pas de distribution particulière mais vont construire une estimation de la fonction densité de probabilité (pdf), c'est notamment le cas de l'*estimation de densité par noyau (KDE)* ou des méthodes reposant sur la *construction d'histogrammes*, mais également d'approches plus récentes issues du DL comme les Variational Auto-Encoders [Kingma 2014] ou les Normalizing Flows [Rezende 2015].

Méthodes basées sur la distance Les méthodes basées sur la distance font l'hypothèse que les instances normales sont proches les unes des autres, tandis que

les anomalies sont éloignées des autres instances. Elles nécessitent donc le choix d'une distance.

Ces méthodes reposent pour la plupart sur l'étude des voisinages des instances et trouvent leur origine dans la définition de [Knorr 1998] d'une (k, R)-anomalie comme une instance n'ayant pas plus de k voisins dans un rayon R autour d'elle. Cette définition se généralise ensuite avec les k plus proches voisins (kNN) en fixant un seuil R sur la distance au k-ième plus proche voisin (k^{th}NN) ou en étudiant la distance moyenne aux kNN.

Méthodes basées sur la densité locale Ces méthodes reposent sur le calcul de métriques mesurant la densité locale, c'est-à-dire la répartition des instances dans le voisinage autour d'une instance, et la comparant à celle autour de ses voisins. L'hypothèse émise est que les instances normales auront une densité locale similaire à celle de leurs voisins, tandis que les anomalies auront une densité locale bien plus faible que leurs plus proches voisins. La métrique la plus connue est le *Local Outlier Factor (LOF)* [Breunig 2000], mais on peut également citer le *Connectivity-based Outlier Factor (COF)* [Tang 2002], l'*INFLuenced Outlierness (INFLO)* [Jin 2006] et le *Multi-granularity DEviation Factor (MDEF)* [Papadimitriou 2003].

Méthodes reposant sur un clustering Les méthodes reposant sur un clustering sont parfois regroupées avec les méthodes basées sur la distance ; en effet, le clustering consiste à regrouper les instances dans l'espace en groupes ou clusters, et le critère pour réaliser ce regroupement est souvent la distance entre les instances.

Le problème du clustering est un champ de recherche à lui seul, et il existe donc de nombreuses méthodes pour le traiter. L'adaptation du clustering à la détection d'anomalies est généralement réalisée à travers l'une des hypothèses suivantes :

- les instances normales appartiennent à des clusters tandis que les anomalies n'appartiennent à aucun cluster ; cette hypothèse suppose que l'approche de clustering soit en mesure de rejeter des instances,
- les instances normales sont proches du centroïde de cluster le plus proche tandis que les anomalies en sont éloignées ; cette hypothèse nécessite de calculer l'emplacement des centroïdes, barycentres des clusters, pour pouvoir étudier l'anormalité des instances,
- les instances normales appartiennent à des clusters denses, où la densité est définie par le nombre d'éléments dans le cluster que divise le volume du cluster, tandis que les anomalies appartiennent à des clusters épars ; il faut cette fois que l'approche de clustering puisse regrouper les anomalies dans des clusters.

On peut également citer la méthode des *forêts d'isolation* comme approche en marge entre les méthodes basées distance et les méthodes de clustering. En effet, il s'agit d'une adaptation d'une approche de clustering réalisant des partitionnements aléatoires de l'espace et qui fait une hypothèse différente des précédentes, à savoir que les anomalies sont les instances qui seront en moyenne les plus rapides à isoler.

Méthodes de classification Les méthodes de classification font l'hypothèse qu'un classifieur peut apprendre à séparer les instances normales des anomalies. Tout comme pour le domaine du clustering, le problème de la classification est un domaine de recherche à part entière. Néanmoins, la plupart de ses méthodes sont supervisées.

Il existe tout de même des approches de classification à une classe qui peuvent être utilisées dans un contexte semi-supervisé. Dans la catégorie des machines à vecteurs de support, on peut notamment citer le *One-Class SVM* [Schölkopf 2001] comme célèbre méthode pour la détection d'anomalies. Les approches de classification à une classe ont également été renforcées par le développement du DL [Ruff 2018].

Méthodes par reconstruction L'objectif des méthodes par reconstruction est de générer un encodeur et un décodeur, se comportant donc comme un modèle de reconstruction lorsqu'ils sont couplés. L'encodeur réalise un encodage de l'instance en entrée dans un espace de plus faible dimension, et le décodeur permet de reconstruire l'instance dans l'espace initial.

L'hypothèse de ces méthodes est que, les instances normales étant plus nombreuses dans le jeu de données d'entraînement et étant générées par un processus différent de celui des anomalies, le modèle appris aura de meilleures capacités à reconstruire une instance normale qu'une anomalie. Ainsi, l'erreur de reconstruction peut être utilisée comme score d'anomalie.

Les approches les plus connues dans cette catégorie sont l'*analyse en composantes principales* [Hawkins 1974] et les *auto-encodeurs* [Sakurada 2014].

Méthodes ensemblistes Les méthodes ensemblistes sont des approches particulières qui combinent plusieurs modèles et qui peuvent avoir deux objectifs :

— augmenter la robustesse du modèle général à travers la combinaison de plusieurs modèles ; les modèles combinés sont ici générés par des méthodes similaires mais avec des paramètres variables,
— éviter le choix d'une méthode en combinant des modèles générés par des méthodes différentes.

Les approches ensemblistes sont principalement utilisées dans les domaines de la classification et du clustering, mais sont plus difficilement applicables à la détection d'anomalies [Wang 2019]. Cependant, l'approche ensembliste promettant une plus grande robustesse des méthodes, elle peut être vue comme une perspective d'amélioration intéressante pour les méthodes des autres catégories.

1.2.3 La détection d'anomalies dans les réseaux de capteurs

1.2.3.1 Intérêt

Revenons maintenant au contexte des WSNs. La Section 1.1.2 a affirmé que leur application nécessitait des mesures précises pour étudier le comportement des sys-

tèmes qu'ils observent, et que ces mesures étaient réalisées par des noeuds capteurs avec des capacités de calcul, mémoire et énergétique limitées.

Aussi, la détection d'anomalies peut être appliquée aux WSNs avec l'objectif de détecter deux types d'anomalies :

— des *évènements d'intérêt*, qui peuvent correspondre par exemple à un nouveau comportement du système étudié ou à une défaillance, mais qui auront dans tous les cas un intérêt majeur dans l'étude des systèmes sur lesquels les WSNs sont déployés,

— des *erreurs*, qui correspondent cette fois à des mesures erronées, pouvant être liées à un défaut sur un noeud capteur ou à une défaillance dans son fonctionnement dû, par exemple, au vieillissement de sa batterie.

La différence entre ces deux types d'anomalies tient principalement à leur portée ; les erreurs n'impactent généralement que les mesures d'un noeud tandis que les évènements d'intérêt peuvent impacter toute une région.

Dans la littérature, on rencontre souvent un troisième type d'anomalies qui correspond aux attaques malveillantes sur le réseau, avec un risque plus ou moins élevé selon le cas d'application. Ces anomalies sont d'origine humaine et peuvent prendre différentes formes selon le type d'attaque, avec l'objectif de ne pas être détectées ou assimilées aux deux types d'anomalies cités. Aussi, elles ne seront pas considérées ici comme un type d'anomalies différent.

La détection de ces anomalies, mais également leur distinction et leur diagnostic, sont d'un intérêt capital pour l'analyse des systèmes à travers les WSNs et dans un contexte IIoT. Il s'agit d'un objectif de cette thèse.

1.2.3.2 Spécificités

Au sein des WSNs, les données sont générées sous la forme de flux continus. De plus, les attributs peuvent être de types variés. Les spécificités des WSNs forcent également plusieurs spécificités dans les méthodes de détection d'anomalies dédiées à leur étude :

— une *opération "en ligne"* ; les données étant générées sous la forme de flux, l'apprentissage ne peut pas être réalisé sur l'intégralité du jeu de données, il doit pouvoir être fait de manière incrémentale pour que le modèle s'adapte aux nouvelles instances,

— la possibilité d'une *analyse distribuée* ; les données étant générées de manière distribuée, et pour éviter un sur-coût de communications, il est également préférable de les analyser de manière distribuée,

— la *légèreté des méthodes et des modèles* ; pour tenir compte des capacités limitées des noeuds dans le réseau, et pour maintenir une analyse distribuée, les méthodes doivent être légères à exécuter et les modèles doivent occuper une place minimale en mémoire,

— la prise en compte de *dépendances* ; on peut distinguer trois types de dépendances dans les WSNs : une *dépendance temporelle* entre les différentes mesures relevées par chaque noeud, une *dépendance spatiale* entre les me-

sures relevées par les différents noeuds et une *dépendance d'attributs* entre les différentes variables.

Les spécificités listées, qui auront une influence sur les méthodes utilisées pour la détection d'anomalies dans les WSNs, sont en réalité fortement liées aux spécificités des flux de données.

1.3 Introduction du modèle des flux de données

Cette section présente les flux de données, qui peuvent être utilisés pour modéliser les jeux de données générés au sein des WSNs, et qui ont reçu beaucoup d'attention dans le domaine de la détection d'anomalies comme ont pu le montrer nos travaux [Ducharlet 2022].

1.3.1 Qu'est-ce qu'un flux de données ?

Le développement rapide de l'informatique a permis de multiplier les sources de données. De nombreuses applications telles que le suivi d'activité web, le suivi de l'évolution des parts de marché en finance, la surveillance du trafic réseau ou les WSNs, génèrent des données de manière continue sous forme de flux en temps réel appelés flux de données.

Ces flux de données sont horodatés, leur vitesse d'arrivée est généralement élevée, et dans certains cas n'est pas connue à l'avance. Enfin, la distribution dans ces flux de données peut changer. On parle alors de cadre "en ligne" par opposition au cadre "hors ligne" où la distribution est stationnaire et où le modèle n'a pas besoin d'évoluer au cours du temps.

Les travaux autour des flux de données se sont multipliés depuis le début des années 2000 [Abadi 2003, Motwani 2003], notamment dans la conception de bases de données pour faciliter leur stockage et optimiser les requêtes. Leurs diverses spécificités rendent leur analyse complexe, ce qui en fait un sujet d'intérêt prioritaire dans le domaine de la fouille de données, et particulièrement en détection d'anomalies où les méthodes dédiées aux flux de données se multiplient ces dernières années.

Définition 2 *Un flux de données est un jeu de données* $\mathcal{D} := \{d_t, t \in \mathbb{N}\}$ *de taille infinie où chaque élément* d_t *correspond à un couple* $d_t := (\tau_t, \mathbf{x}_t)$ *d'une valeur p-variée* $\mathbf{x}_t$ *horodatée par une date unique* τ_t*. Ce flux est généré par une source avec une périodicité pouvant, selon le cadre d'application, ne pas être fixe ; pour* $i \neq j$ *et* $i, j > 0$*, on peut avoir* $\tau_i - \tau_{i-1} \neq \tau_j - \tau_{j-1}$*.*

Cependant, on ne dispose généralement pas de toutes les instances de $\mathcal{D}$ depuis $t = 0$, mais seulement à partir de $t = \alpha \geq 1$, et on ne connaît pas les instances qui seront mesurées dans le futur. On travaille donc avec des flux de données partiels.

Définition 3 *Un flux de données partiel est défini, à chaque instant* t*, comme* $\mathcal{D}_{\alpha,t} := \{d_i, \alpha \leq i \leq t\}$*.*

En détection d'anomalies, pour évaluer l'anormalité de l'instance $\mathbf{x_t}$, on utilise donc le jeu de données $\mathcal{D}_{\alpha,t-1}$. Par abus de langage, on parlera toujours de flux de données pour désigner des flux de données partiels.

Les WSNs font partie des sources générant des flux de données. Spécifiquement, on peut considérer que chaque noeud capteur génère son propre flux de données, en effet chaque noeud capteur relève, à différents instants τ_t, un ensemble de $p \geq 1$ mesures $\mathbf{x_t}$. Ainsi, traiter le cas des WSNs revient à analyser un ensemble de flux de données.

1.3.2 Spécificités des flux de données

1.3.2.1 Liste des spécificités

En introduisant les différentes problématiques de recherches soulevées par l'application de la détection d'anomalies aux flux de données, [Sadik 2014] fournit une liste de leurs spécificités :

- *caractère éphémère* : chaque instance d_t porte une information dont l'importance décroît avec le temps, détecter une anomalie trop tard n'a que peu d'intérêt, et les instances peuvent même n'exister que pendant une durée limitée ; elles doivent donc être traitées aussitôt qu'elles sont mesurées,
- *temporalité* : chaque instance d_t est associée à une date τ_t qui doit être considérée, que ce soit comme un attribut ou en étudiant l'ordre de génération des instances ; dans tous les cas, étudier une instance dans son contexte temporel a un intérêt,
- *infinité* : les mesures pouvant être générées en continu, $\mathcal{D}$ est de taille infinie, on ne peut pas stocker en mémoire les flux, même partiels, indéfiniment ; les méthodes ne doivent travailler qu'avec un modèle qui résume le flux de données,
- *vitesse de génération* : elle peut être fixe comme variable, mais les instances devant être traitées dès qu'elles sont générées, le temps d'exécution de la méthode de détection d'anomalies doit être court, et potentiellement adaptable si la vitesse de génération varie rapidement,
- *non-stationnarité* : dans la plupart des cas, la distribution n'est pas stationnaire, elle peut évoluer au cours du temps ; les méthodes faisant l'hypothèse d'une distribution fixe ne sont donc pas applicables,
- *incertitude* : dans certains cas d'application, et notamment dans celui des WSNs, les mesures peuvent être perturbées par l'environnement, auquel cas les méthodes doivent pouvoir être robustes au bruit qui en découle, et il peut également y avoir des mesures manquantes ou arrivant en retard, qui doivent tout de même être analysées dans leur contexte temporel,
- *multi-dimensionnalité* : celle-ci a deux aspects dans le contexte des WSNs, d'abord au niveau des instances p-variées lorsque $p > 1$, mais aussi avec la multitude de noeuds capteurs ; il est intéressant de pouvoir réaliser une étude multi-variée en tenant compte de ces deux aspects,

— *caractère embarqué* : ajoutons ici une dernière spécificité pour le cas des WSNs avec le calcul embarqué dans les noeuds en périphérie du réseau ; les capacités de calcul et mémoire étant limitées, les méthodes doivent être légères.

Ainsi, les principales caractéristiques souhaitées pour une méthode de détection d'anomalies appliquée au traitement des flux de données, en particulier dans les réseaux de capteurs, sont :

— sa rapidité d'exécution,
— sa capacité à étudier les instances dans leur contexte temporel,
— la légèreté de son modèle, qui ne doit pas nécessiter de stocker toutes les instances en mémoire,
— sa capacité à apprendre en continu,
— sa robustesse
— et sa capacité à tenir compte de la multi-dimensionnalité du problème.

De plus, le contexte des flux de données *impose un contexte non supervisé*. En effet, le fait qu'on ne dispose que d'un flux partiel à chaque instant, couplé au caractère non-stationnaire de la distribution, rendrait la labellisation rapidement obsolète. Dans le meilleur des cas, une approche hybride pourrait être envisageable, avec un premier apprentissage supervisé ou semi-supervisé, puis une évolution du modèle non supervisée.

1.3.2.2 Différence avec les séries temporelles

Les séries temporelles et les flux de données sont souvent assimilées. L'article de [Duraj 2021] réalise notamment une étude des méthodes de détection d'anomalies dans les séries temporelles en définissant les jeux de données utilisés comme des séries temporelles.

Nous différencions ici l'étude des flux de données de celle des séries temporelles dans leur rapport au temps. En effet, les séries temporelles sont décrites à travers des saisonnalités et des tendances, et leur étude repose sur la modélisation de ces caractéristiques dans des modèles de prédiction des instances futures à partir des instances passées. Les tendances et saisonnalités, bien que provoquant des changements dans la distribution des données, induisent une régularité qui n'est pas suffisante pour la modélisation du comportement des flux de données. Enfin, ces méthodes réalisent généralement un apprentissage en une seule passe pour générer le modèle qui n'évolue plus avec les nouvelles instances générées.

1.3.2.3 Techniques de fenêtrage

Une solution simple pour satisfaire un bon nombre des spécificités des flux de données est l'utilisation de techniques de fenêtrage. Des fenêtres glissantes sont utilisées afin de ne conserver en permanence qu'une sous-partie du flux de données ne contenant que des points récents, limitant l'occupation mémoire et permettant de toujours traiter les nouvelles instances dans leur contexte temporel.

Il existe plusieurs approches pour le fenêtrage [Salehi 2018] :

— le *fenêtrage par point de repère* fixe un point de repère comme début de la fenêtre et contient toutes les instances de ce point de repère jusqu'à la date courante ; le jeu de données consiste alors simplement en un flux de données partiel $\mathcal{D}_{\alpha,t}$ avec α fixe, et il est nécessaire de modifier ce α lorsque la taille du jeu de données devient trop grande,

— le fenêtrage glissant fixe la taille de la fenêtre, que ce soit dans le temps ($\tau_t - \tau_\alpha$ fixe) ou en nombre d'éléments contenus dans $\mathcal{D}_{\alpha,t}$, on a donc un α dépendant du temps et la fenêtre glisse en continu, ce qui permet de limiter la taille du jeu de données,

— le *fenêtrage amorti* associe à chaque instance un poids selon son ancienneté, ainsi les points les plus récents auront un poids plus élevé que les plus anciens ; cette approche évite d'avoir à retirer les instances qui sortent de la fenêtre,

— le *fenêtrage adaptatif* est similaire à un fenêtrage glissant mais la taille de la fenêtre n'est pas fixe, elle dépend de la vitesse à laquelle les données évoluent ; ainsi, la fenêtre sera grande si la distribution est stable et petite si la distribution évolue rapidement.

La solution du fenêtrage, peu importe l'approche retenue, peut cependant s'avérer coûteuse en temps de traitement. En effet, les méthodes de détection d'anomalies applicables aux jeux de données stationnaires ne permettent pas toutes de mettre à jour le modèle appris avec de nouveaux points. Dans ce cas, il faut réaliser un nouvel apprentissage complet à chaque fois que la fenêtre change, ce qui s'avère très lourd.

Les méthodes applicables à la détection d'anomalies dans les flux de données avec une technique de fenêtrage sont donc des méthodes qui permettent de mettre à jour le modèle rapidement à chaque fois que la fenêtre est actualisée. Des méthodes qui permettent une représentation du jeu de données complet sous une forme réduite (une matrice dont la taille n'est pas dépendante du nombre d'instances dans le jeu de données par exemple) sont également utilisables.

1.4 Question de l'évaluation dans un contexte non supervisé

La dernière thématique abordée avant d'introduire les objectifs de cette thèse concerne la difficulté d'évaluation des performances de la détection d'anomalies dans un contexte non supervisé, imposé par le cadre des flux de données.

1.4.1 Description du problème

1.4.1.1 Évaluation d'un modèle de détection d'anomalies

Comme décrit dans la Section 1.2.2.1, une méthode de détection d'anomalies peut produire deux types de résultats en sortie : un score ou un label. Dans la

majorité des cas, et en particulier pour les approches non supervisées, la sortie est un score, et on peut déduire un label pour les instances en fixant un seuil sur le score.

Évaluer la qualité d'un modèle de détection d'anomalies revient à étudier ses performances, qui peuvent se décliner selon deux types : les performances en détection et les performances algorithmiques.

Performances en détection Les performances en détection ont pour objectif d'étudier à quel point le modèle est satisfaisant dans sa capacité à détecter des anomalies. Intuitivement, elles peuvent être évaluées de la même manière qu'un problème de classification avec deux classes : la classe normale et la classe anormale.

Pour évaluer une classification à deux classes, l'approche classique est l'utilisation de matrices de confusion, comme décrite dans le Tableau 1.2.

		Classe prédite	
		Anomalie	Normal
Classe réelle	Anomalie	Vrais Positifs (VP)	Faux Négatifs (FN)
	Normal	Faux Positifs (FP)	Vrais Négatifs (VN)

TABLE 1.2 – Contenu d'une matrice de confusion

On peut déduire de cette matrice de confusion un certain nombre de métriques utiles pour évaluer une classification :

- le rappel (ou taux de vrais positifs) $R = \frac{VP}{VP+FN}$, qui correspond au taux d'anomalies du jeu de données correctement classées comme anormales,
- la précision $P = \frac{VP}{VP+FP}$, qui correspond au pourcentage d'anomalies réelles parmi toutes les instances classées comme anormales,
- la spécificité (ou sélectivité ou taux de vrais négatifs) $S = \frac{VN}{VN+FP}$, qui correspond au taux d'instances normales du jeu de données correctement classées comme normales, peu utilisée en détection d'anomalies où la classe positive (les anomalies) est plus importante que la classe négative ; on lui préfère donc le rappel,
- le taux de faux positifs (ou de fausses alarmes) $FPR = 1 - S$, qui correspond au taux d'instances normales du jeu de données considérées comme anormales, il est important de le considérer en détection d'anomalies car il augmente lorsqu'on augmente le rappel,
- l'exactitude $Acc = \frac{VP+VN}{VP+VN+FP+FN}$, qui correspond au taux de bonnes prédictions sur l'ensemble du jeu de données. Cependant, l'exactitude n'a que peu d'intérêt dans le cas de la détection d'anomalies où la classe positive est généralement largement moins représentée en proportion, et pourtant plus importante. On remplace donc l'exactitude par l'exactitude équilibrée, calculée comme la moyenne du rappel et de la spécificité $Acc_{eq} = \frac{R+S}{2}$, pour donner autant de poids à la bonne classification d'une anomalie qu'à l'a bonne classification d'une instance normale,

— la F-mesure, calculée comme la moyenne harmonique du rappel et de la précision $F_{measure} = 2 \times \frac{R \times P}{R+P}$, combine ces deux métriques et accorde donc plus d'importance à la bonne classification des anomalies, ce qui la rend pertinente en détection d'anomalies. Cependant, cette métrique dépend du taux d'anomalies dans le jeu de données ce qui rend la comparaison de la F-mesure entre plusieurs cas d'application difficile.

Performances algorithmiques En plus de la performance en détection, il est également important d'étudier les performances algorithmiques des algorithmes, ce qui revient à étudier leur temps d'exécution (en phase d'apprentissage et en phase d'évaluation) mais également l'occupation mémoire des modèles.

L'étude de la performance algorithmique est réalisée pour évaluer les méthodes dans le cas d'applications avec des contraintes en ressources [Domingues 2018]. C'est notamment le cas dans le cadre des WSNs et des flux de données, où les méthodes se doivent d'être légères.

1.4.1.2 Limites du cadre non supervisé

Si la performance algorithmique n'est pas dépendante du cadre d'application, c'est en revanche le cas pour la performance en détection. En effet, dans le cas supervisé, une instance mal classée correspond à une erreur, alors qu'en détection d'anomalies non supervisée, s'il n'y a que 10 anomalies dans un jeu de données comprenant des millions d'instances et qu'elles sont classées parmi les 15 instances avec un score d'anomalie le plus élevé, on peut considérer qu'il s'agit d'un résultat satisfaisant [Goldstein 2016].

De plus, la création d'une matrice de confusion nécessite de disposer de labels pour la classe réelle, et le cadre non supervisé implique l'indisponibilité de ces labels. L'approche classique consiste alors à réaliser un "benchmark" en utilisant plusieurs jeux de données labellisés pour évaluer les performances globales d'une méthode, c'est ce qu'on appelle une *évaluation externe*. L'inconvénient de cette approche est que les performances d'une méthode sur un jeu de données ne garantissent pas qu'elles seront équivalentes sur un autre jeu de données, il faut donc que le benchmark comporte un ensemble suffisamment représentatif de jeux de données, ce qui est difficilement vérifiable.

Une seconde approche consiste à réaliser une *évaluation interne* (ou *agnostique*) qui ne repose pas sur la présence de labels mais uniquement sur les données elles-mêmes. Ces approches sont rares, et elles nécessitent des hypothèses fortes sur les anomalies ou les fonctions de score des méthodes. Ainsi, elles privilégient naturellement certaines méthodes par rapport à d'autres.

De manière générale, l'évaluation de la performance en détection dans le cadre de la détection d'anomalies non supervisée n'a que peu de sens et ne peut être réalisée que sous certains compromis ou certaines hypothèses.

1.4.2 L'approche usuelle : évaluation externe

L'approche la plus fréquemment utilisée pour évaluer des méthodes de détection d'anomalies dans un cadre non supervisé consiste à (1) générer un benchmark à partir de plusieurs jeux de données labellisés et (2) choisir une ou plusieurs métriques adaptées à ces méthodes et reposant sur des labels.

1.4.2.1 Construction du benchmark

Il existe trois stratégies pour construire un benchmark pour la détection d'anomalies [Ruff 2021] :

— choisir des jeux de données pour la classification, sélectionner une ou plusieurs classes pour représenter la normalité et une ou plusieurs classes pour représenter les anomalies. Dans le domaine de la classification, le déséquilibre des classes est généralement moins important que dans le domaine de la détection d'anomalies ; il est donc souvent nécessaire de réaliser un sous-échantillonnage des classes représentant les anomalies pour simuler ce déséquilibre,

— générer des jeux de données synthétiques avec une ou plusieurs distributions normales et des anomalies, ou générer des anomalies synthétiquement dans des jeux de données réels ne contenant que des instances normales,

— choisir des jeux de données réels contenant des anomalies labellisées par un expert du domaine qui peut ajouter des informations sur les causes des anomalies.

Bien que la première stratégie soit adaptée aux définitions de l'anomalie fournies dans la Section 1.2.1.1, elle manque de robustesse ; en effet, on peut obtenir des résultats très différents en faisant varier les classes choisies comme normales et anormales et en réalisant un sous-échantillonnage différent [Campos 2016].

La seconde stratégie souffre des mêmes avantages et inconvénients que la première ; bien qu'elle soit adaptée aux définitions fournies pour le concept d'anomalie, ses résultats sont très dépendants du processus ayant permis de générer les anomalies.

La troisième stratégie est la meilleure d'un point de vue industriel, mais a un coût plus important puisqu'elle demande un effort de labellisation de la part d'un expert. De plus, cette approche est surtout pertinente si les jeux de données utilisés pour l'évaluation sont similaires aux jeux de données réels sur lesquels la méthode sera appliquée de manière non supervisée. Or c'est justement parce que cet effort de labellisation serait trop coûteux que le cadre non supervisé est généralement choisi.

1.4.2.2 Choix des métriques

On retrouve souvent les mêmes métriques dans les études comparatives de méthodes de détection d'anomalies dans un cadre non supervisé. Comme expliqué par [Goldstein 2016], ces métriques ne visent pas à récompenser la classification exacte mais le classement des instances selon leur score d'anomalies. En effet, dans le cas

de la détection d'anomalies non supervisé, les modèles attribuent un score d'anomalie à chaque instance, et il est donc possible de classer les instances de la plus anormale (score élevé) à la plus normale (score faible).

P@k et R@k L'une des métriques les plus fréquentes est la précision à k (P@k), calculée comme le taux d'anomalies réelles parmi les k instances ayant le score le plus élevé. Cette approche est cependant très critiquée car trop sensible au choix du paramètre k [Schubert 2012, Zimek 2018]. En effet, si on reprend l'exemple d'un jeu de données de plusieurs millions d'instances dont 10 anomalies, la P@10 d'une méthode qui classera les 10 anomalies aux places 11-20 sera la même que celle d'une méthode qui les classera aux dernières places. A l'opposé, la P@20 d'une méthode les classant aux places 1-10 sera également la même que celle de la méthode les classant aux places 11-20.

Moins fréquent, le rappel à k (R@k) peut être utilisé lorsqu'on cherche à maximiser le nombre de vrais positifs pour un nombre de fausses alarmes limité, il correspond à la proportion d'anomalies réelles pour k faux positifs [Ruff 2021].

Ces deux métriques sont surtout utiles lorsque des contraintes sont spécifiées et qu'un coût associé à l'erreur de manquer une anomalie ou de lever une fausse alarme est fourni, ce qui est rarement le cas en pratique.

AUROC et AUPRC/AP La métrique la plus populaire aujourd'hui pour l'évaluation de la détection d'anomalies non supervisée est l'aire sous la courbe ROC (AUROC). La courbe ROC est générée en traçant le rappel R en fonction du taux de faux positifs FPR pour les différentes valeurs de seuil sur le score d'anomalies possibles. L'AUROC est ensuite calculée comme l'aire sous la courbe. Sa popularité est notamment due au fait qu'elle puisse être facilement interprétée comme la probabilité qu'une anomalie choisie aléatoirement ait un score supérieur à une instance normale choisie aléatoirement. Elle a également une valeur de référence fixe de 0.5 ce qui la rend facilement comparable d'un cas d'application à l'autre. Cependant, l'AUROC est connue pour donner des résultats optimistes dans le cas d'un jeu de données très déséquilibré.

Au contraire, l'aire sous la courbe rappel-précision (AUPRC), où la courbe rappel-précision (PR) est tracée en opposant la précision P au rappel R, offre une meilleure robustesse au déséquilibre des classes mais est moins interprétable ; en effet, la valeur de référence dépend cette fois du taux d'anomalies dans le jeu de données, et les valeurs de cette métriques ne peuvent donc pas être facilement comparées d'un cas d'application à un autre. Aussi, l'AUPRC peut être estimée avec la précision moyenne (AP) comme décrit par [Boyd 2013] (en fixant dans leur article $\pi = \frac{n}{n+m}$).

L'avantage principal de ces deux métriques est qu'elles ne nécessitent pas de fixer un seuil sur le score d'anomalies ou le paramètre k mais sont calculées sur l'étendue du spectre de scores d'anomalies possibles.

Autres solutions moins communes Une approche permettant de régler la dépendance à k de la P@k est d'utiliser la P@k moyenne sur un intervalle de valeurs de k, ce qui revient au calcul de la précision moyenne à k (AP@k). [Campos 2016] propose également un ajustement de la P@k et de l'AP@k qui permet de les rendre comparable entre différents cas d'application.

[Schubert 2012] propose également différentes métriques s'appuyant à la fois sur le classement et sur la valeur du score d'anomalies pour améliorer les résultats obtenus avec les métriques P@k et AUROC. Aussi, afin de pousser le développement de l'approche ensembliste pour la détection d'anomalies, les métriques proposées sont pensées pour être applicables dans une telle approche.

Enfin, il existe des métriques bien plus spécifiques reposant sur des benchmarks tout aussi spécifiques. Dans le cas de l'étude des séries temporelles, le cas le plus connu et le Numenta Anomaly Benchmark (NAB) proposant un score qui vise à favoriser la détection d'une anomalie quelques instants avant son occurrence [Lavin 2015]. Il s'agit sans doute de l'approche d'évaluation externe la plus avancée à ce jour, mais elle reste dépendante de la variété de jeux de données à disposition et de la qualité de la labellisation, bien qu'une réglementation sur la labellisation soit en place pour filtrer les jeux de données pouvant contribuer au NAB.

1.4.3 L'approche agnostique : évaluation interne

L'approche agnostique a reçu bien moins d'attention ces dernières années. A ma connaissance, les trois métriques mentionnées ci-après sont les seules à proposer une évaluation interne. Ces métriques nécessitent cependant de faire des compromis importants ou des hypothèses fortes sur les anomalies ou les scores d'anomalie.

IREOS IREOS a été proposée par [Marques 2015] en s'inspirant de l'évaluation interne pour le clustering, également non labellisé, où des indices internes sont utilisés pour évaluer la qualité du regroupement réalisé.

Cette métrique fait l'hypothèse qu'une anomalie doit être facilement séparable du reste des données, et elle utilise un classifieur binaire pour étudier la séparabilité des instances considérées comme anormales. Cependant, pour être calculée, il est nécessaire de générer un certain nombre de classifieurs pour pouvoir séparer toutes les anomalies et calculer une distance de chaque anomalie à chaque classifieur, ce qui revient à une complexité algorithmique lourde pour l'évaluation

Il est également important de noter que, de par l'hypothèse sous-jacente à IREOS, une méthode reposant sur le principe de séparation des anomalies, comme les forêts d'isolation, obtiendra naturellement de meilleurs résultats avec cette métrique.

EM et MV L'utilisation des courbes Excess-Mass (EM) et Mass-Volume (MV) est proposée par [Goix 2016] pour remplacer les courbes ROC et PR, avec l'avantage de ne pas nécessiter de labels.

Ces métriques font l'hypothèse que la fonction de score doit suivre la même évolution que la densité de probabilité de la distribution ayant généré les instances. Les courbes, et les aires sous les courbes, sont estimées via une méthode de Monte-Carlo. L'article met en évidence le rapprochement de la classification des méthodes selon ces métriques avec celle obtenue avec l'AUROC et l'AUPRC.

Tout comme pour IREOS, cette approche favorise les méthodes réalisant une hypothèse similaire comme les méthodes statistiques reposant sur une estimation de la densité.

CC-Eval Enfin, nos travaux ont également mené à la génération d'une métrique pour l'évaluation agnostique. CC-Eval, proposée dans [Ducharlet 2020], n'évalue pas la qualité de la détection mais sa cohérence. Pour se faire, un classifieur est entraîné avec les prédictions réalisées par les méthodes de détection d'anomalies. Pour de nouvelles instances, la prédiction réalisée par la méthode est comparée à celle réalisée par le classifieur. La cohérence est mesurée par la similitude des deux prédictions.

Cette approche seule n'est cependant pas suffisante pour qualifier de la qualité d'un modèle et ne peut qu'être utilisée pour s'assurer de sa cohérence. Elle est également grandement dépendante du classifieur choisi et de sa proximité à la méthode de détection d'anomalies ; un SVM comme classifieur aura plus de facilités à retrouver le modèle appris par un One-Class SVM.

1.5 Problématique, objectifs et organisation

Cette section formalise la problématique industrielle et les objectifs et verrous de cette thèse, en rapport avec les observations réalisées dans les sections précédentes. Elle se conclut en présentant l'organisation du manuscrit pour les chapitres suivants.

1.5.1 Problématique industrielle

Comme mentionné en introduction de ce premier chapitre, la problématique industrielle est l'analyse de la fiabilité des données issues de WSNs dans un vaste panel de cas d'application industriels et en amont d'une phase de diagnostic plus poussée.

En effet, dans ces travaux, les WSNs sont utilisés au sein d'un service de GMAO pour suivre le fonctionnement de systèmes dans des cas d'application variés provenant de clients. L'analyse de la fiabilité des données générées pas ces WSNs a lieu comme processus de fond ; il ne doit pas solliciter l'intervention d'un expert métier du côté client et il serait coûteux de demander à un opérateur d'adapter le processus spécifiquement à chaque cas d'application. Ainsi, en suivant le besoin industriel, il est nécessaire de développer une solution automatisable et agnostique.

Cette solution a tout de même pour vocation d'être positionnée en amont d'une phase de diagnostic plus poussée qui pourra, elle, intégrer des connaissances d'un

expert métier côté client, à travers une phase d'interaction. Cette phase ne fait cependant pas partie du périmètre de cette thèse.

La problématique d'une solution agnostique mène naturellement vers le choix d'une approche basée données et vers le domaine de la détection d'anomalies. En suivant le schéma présenté dans la Section 1.2.2.1, la détection d'anomalies peut être formulée ainsi dans cette problématique industrielle :

— nature des données : les données considérées seront sous la forme d'un ensemble de flux de données p-variés avec des valeurs quantitatives continues, et on se restreint à de faibles dimensions ($2 \leq p \leq 4$), en revanche le nombre de noeuds capteurs dans les WSNs, et donc le nombre de flux de données, n'est pas limité, et plusieurs types de dépendances peuvent exister : une dépendance temporelle (entre les mesures), une dépendance spatiale (entre les différents flux) et une dépendance d'attributs (entre les variables),
— types d'anomalies : on ne considère que les anomalies des WSNs, à savoir les erreurs et les évènements d'intérêt (sans considérer les attaques malveillantes) ; la détection de ces anomalies peut dans tous les cas se ramener à la détection d'anomalies ponctuelles, avec l'intuition qu'on pourra les identifier en s'appuyant sur les différentes dépendances,
— labellisation : la nature des WSNs, des flux de données et le besoin d'une solution automatisable et agnostique conduisent irrémédiablement vers l'absence de labels et une approche non supervisée,
— résultat attendu : les méthodes appliquées doivent fournir un score d'anomalie, celui-ci portant plus d'informations pour le diagnostic en aval qu'un label seul.

Cette formulation de la détection d'anomalies entraîne la question du choix des méthodes. Comme décrit au long de la Section 1.4, il est difficile d'évaluer les méthodes de détection d'anomalies dans un contexte non supervisé. Cette observation est renforcée par la problématique industrielle présentée, où les cas d'application sont variés et ne sont pas connus à l'avance. Dans ces conditions, le choix d'une méthode est difficile.

1.5.2 Exemple de cas d'application

L'exemple de cas d'application ayant motivé les travaux de cette thèse est un ensemble de convoyeurs de bagages aéroportuaires. Des capteurs sont installés sur plusieurs convoyeurs pour étudier leur fonctionnement, comme illustré par les photos en Figure 1.2.

Plusieurs grandeurs sont relevées par les capteurs, parmi lesquelles la vitesse du tapis, l'intensité du moteur, la température du moteur et la température d'huile. Ces différentes grandeurs sont fortement liées à l'état de fonctionnement des convoyeurs.

Le jeu de données est décrit pour deux grandeurs, la vitesse et l'intensité, par la Figure 1.3. A l'arrêt, les deux grandeurs prennent des valeurs nulles. Au démarrage, un pic d'intensité peut être brièvement constaté pendant l'accélération du tapis, puis l'intensité et la vitesse se stabilisent. Lors de l'arrêt du convoyeur, l'intensité

FIGURE 1.2 – Photos d'un élément de convoyeur (tapis à gauche et capteurs branchés à droite)

revient rapidement à zéro tandis que la vitesse décroît un peu plus lentement. En cas de bagage lourd, mais aussi dans le cas d'un redémarrage avant arrêt complet, un pic d'intensité peut être constaté avec une vitesse inférieure à la vitesse nominale.

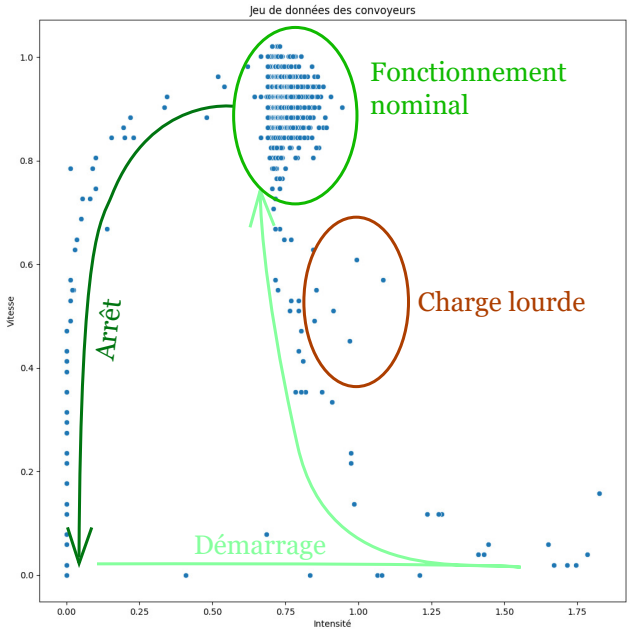

FIGURE 1.3 – Description du jeu de données d'un convoyeur

Bien que ce jeu de données soit simple à comprendre, il est difficilement traitable en non supervisé. En effet, les différents états du convoyeur décrits précédemment ne sont pas aussi représentés dans le jeu de données ; les convoyeurs peuvent être à l'arrêt 70% du temps, et lorsqu'ils sont en fonctionnement les grandeurs prennent majoritairement les valeurs de fonctionnement nominal. Aussi, les valeurs de transition peuvent facilement être considérées comme anormales à cause de leur faible proportion dans le jeu de données, induisant des faux positifs dans le cadre de la détection d'anomalies.

Les travaux décrits dans la suite du manuscrit se voulant génériques, on ne souhaite pas s'appuyer uniquement sur les résultats sur ce jeu de données. De plus, nous ne disposons pas de labels pour ce jeu de données pour permettre l'évaluation des résultats. Il sera tout de même utilisé dans le Chapitre 5 pour illustrer les résultats des méthodes proposées dans un cas d'application industriel.

1.5.3 Objectifs de la thèse et verrous scientifiques

Suite à la formalisation de la problématique industrielle, l'objectif de la thèse et les potentiels verrous sont définis ci-après.

Objectif 1 *Proposer une solution,* automatisable *et* agnostique au contexte industriel*, à la problématique de détection d'anomalies dans les réseaux de capteurs.*

Cette introduction permet déjà d'identifier plusieurs verrous associés à cet objectif, qui seront ensuite affinés à travers une analyse de l'état de l'art.

Verrou 1 *La sélection de méthodes de détection d'anomalies adaptées aux données parmi le large spectre de méthodes non supervisées de l'état de l'art.*

Verrou 2 *La suppression de la phase de paramétrage de la solution, et donc des méthodes qui la composent.*

Verrou 3 *La satisfaction des contraintes des WSNs, et donc des flux de données, par la solution retenue.*

1.5.4 Organisation du manuscrit

Le manuscrit s'organise de la manière suivante :

— le Chapitre 2 propose un état de l'art des méthodes de détection d'anomalies applicables aux WSNs, ce qui comprend donc également les méthodes proposées dans le contexte des flux de données ; cet état de l'art permet de détailler les verrous précédents et justifie le besoin d'un cadre opérationnel facilitant la sélection de méthodes de détection appropriées,

— le Chapitre 3 décrit ce cadre opérationnel sous la forme d'une première contribution permettant de générer un environnement d'étude propice à la détection d'anomalies non supervisée dans les réseaux de capteurs avec une approche d'évaluation associée ; en s'appuyant sur des définitions spécifiques

de l'anomalie, ce cadre opérationnel est ensuite mis en application avec des méthodes reposant sur l'estimation de densité par noyau,
— le Chapitre 4 présente la seconde contribution qui s'inspire de l'approche ensembliste pour permettre l'intégration commune des différentes méthodes au sein du cadre opérationnel proposé dans le Chapitre 3,
— dans le Chapitre 5, une nouvelle méthode de détection d'anomalies pour les flux de données est développée et évaluée ; celle-ci a la particularité d'être facile à paramétrer et de pouvoir être intégrée au cadre opérationnel du Chapitre 3, elle est également évaluée sur le jeu industriel présenté en Section 1.5.2,
— une amélioration sans paramètres de la méthode du Chapitre 5 est par la suite proposée dans le Chapitre 6 ; ses avantages sont démontrés par de premiers résultats,
— enfin, le Chapitre 7 conclut le manuscrit avec une discussion de l'état de résolution des objectifs de cette thèse et des travaux qui pourraient lui faire suite.

1.5.5 Productions dans le cadre de la thèse

Dans le cadre de cette thèse ont été produits plusieurs articles scientifiques :
— un article pour la détection d'anomalies non supervisées dans un cadre d'apprentissage hors ligne, où on dispose de toutes les données d'apprentissage, accompagné d'une approche pour évaluer la cohérence des modèle appris. L'approche de détection d'anomalies, appelée SuMeRI, facilite le paramétrage des méthodes avec un apprentissage itératif sur le jeu de données, et elle permet également l'utilisation de plusieurs méthodes successivement. L'article a été présenté à la conférence IEA/AIE 2020 [Ducharlet 2020] ;
— un état de l'art des méthodes de détection d'anomalies non supervisées dans les flux de données, présenté à la 20ème édition des Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2022) [Ducharlet 2022] ;
— un article de journal présentant les travaux sur l'application de la fonction de Christoffel empirique, issue des domaines de l'optimisation et des polynômes orthogonaux, à la détection d'anomalies dans les flux de données, dont la soumission est en cours.

De plus, trois productions logicielles ont vu le jour au cours de cette thèse et ont été ou seront utilisées au sein des travaux de l'entreprise Carl Berger-Levrault :
— SuMeRI, une solution pour appliquer successivement plusieurs méthodes de détection d'anomalies avec un apprentissage itératif des modèles ;
— ODDS, un framework pour la détection d'anomalies dans les flux de données ;
— WOLF, un cadre opérationnel pour la détection d'anomalies dans les réseaux de capteurs avec une approche d'évaluation basée définition.

Chapitre 2

État de l'art

Avant de proposer une solution à la problématique industrielle identifiée, il est nécessaire de parcourir l'état des solutions existantes dans le domaine de la détection d'anomalies qui répondent au problème d'identification d'erreurs et d'évènements d'intérêt dans les WSNs. En plus des méthodes appliquées directement aux WSNs, les méthodes de détection d'anomalies dans les flux de données, présentées dans un article rattaché à cette thèse [Ducharlet 2022], seront mentionnées.
Ce chapitre état de l'art suit l'itinéraire des différentes catégories d'approches identifiées dans la Section 1.2.2.2 pour réaliser un tour d'horizon complet de l'existant avant de s'achever sur un tableau récapitulatif des méthodes rencontrées, décrites selon une taxonomie nouvelle.

Sommaire

2.1 Préambule

Dans cette première section, je présenterai tout d'abord le périmètre et la structure de cet état de l'art, puis une taxonomie des méthodes de détection d'anomalies pour les WSNs.

2.1.1 Périmètre et structuration de l'état de l'art

2.1.1.1 Périmètre

Rappelons tout d'abord le périmètre de cette étude, qui définit également celui de cet état de l'art. Le choix d'une approche basée données, reposant sur les méthodes issues du domaine de la détection d'anomalies, a été fixé. Aussi, cet état de l'art ne mentionne que ces méthodes.

Par ailleurs, parmi les méthodes de détection d'anomalies, et notamment parce qu'on se restreint dans cette thèse à un problème en faibles dimensions, les méthodes issues du DL ne seront pas étudiées.

Notons également que plusieurs études de la détection d'anomalies dans les WSNs ont vu le jour jusqu'à aujourd'hui [Zhang 2010, Xie 2011, Ayadi 2017]. Dans ces études, en plus des erreurs et évènements d'intérêt qui sont ciblés par notre problématique, un troisième type d'anomalies apparaît ; les attaques malveillantes, ou intrusions. Il a déjà été expliqué que ces anomalies ne seront pas considérées comme un type distinct car, étant d'origine humaine et malicieuse, elles peuvent prendre l'apparence des deux types d'anomalies précédents ou ne pas être détectables. Aussi, les méthodes ciblant en particulier ces anomalies, souvent dans un contexte de sécurité des WSNs, ne seront pas mentionnées. C'est notamment le cas de la majorité des approches présentées dans l'étude de [Xie 2011].

2.1.1.2 Structuration

La structure de cet état de l'art suit la catégorisation des méthodes de détection d'anomalies présentée dans la Section 1.2.2.2. La section suivante étudie successivement ces catégories, et chaque sous-section est dédiée à l'une d'elles. Pour chaque catégorie, on pourra notamment retrouver :

— quelques fondements mathématiques lorsque ce sera jugé nécessaire,
— certains travaux ayant cherché à appliquer des méthodes de cette catégorie à la détection d'anomalies dans les WSNs,
— certaines méthodes, adaptées aux flux de données, qui pourraient être utilisées dans les WSNs,
— quelques avantages et inconvénients de cette catégorie de méthodes.

Nous choisissons ici de séparer les méthodes développées pour les flux de données des méthodes développées pour les WSNs. En effet, bien que les WSNs génèrent des flux de données, les méthodes de détection d'anomalies dans les WSNs présentées dans l'état de l'art ne considèrent pas systématiquement toutes les spécificités des flux de données, comme nous pourrons l'observer dans ce chapitre.

Finalement, la troisième section réalise un récapitulatif de l'état de l'art et conclut ce chapitre.

2.1.2 Taxonomie des méthodes de détection d'anomalies dans les réseaux de capteurs

Cette sous-section présente une taxonomie nouvelle des méthodes de détection d'anomalies appliquées aux WSNs. Elle permettra de les décrire dans la suite de ce chapitre. Pour commencer, rappelons les différentes contraintes des WSNs :

- les données sont générées sous forme de flux, ce qui amène les différentes spécificités des flux de données et impose une opération en ligne des méthodes,
- une génération distribuée des mesures, avec un coût de communication élevé, tend à pousser le développement de méthodes distribuées,
- des capacités de stockage et de calcul limitées forcent l'utilisation de méthodes rapides et des modèles légers,
- trois types de dépendances sont présentent et devraient être étudiées (dépendances temporelles, spatiales et d'attributs).

La Figure 2.1 présente les différents éléments de la taxonomie introduite. Dans la Section 2.3, un tableau décrit les méthodes présentées selon cette taxonomie.

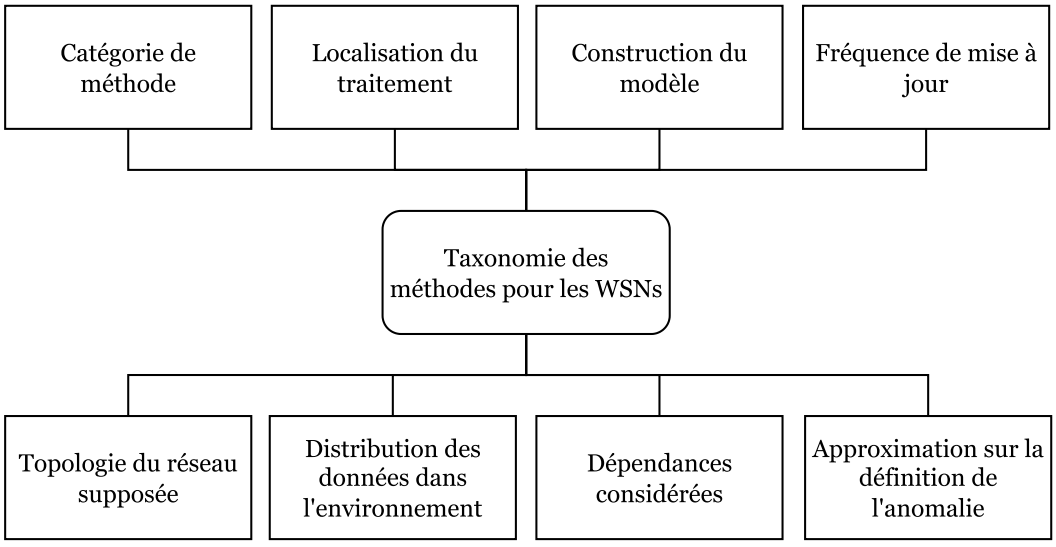

FIGURE 2.1 – Différents éléments de la taxonomie des méthodes de détection d'anomalies pour les réseaux de capteurs

Catégorie de méthode Les méthodes peuvent appartenir à une ou plusieurs des catégories présentées dans le chapitre précédent : les méthodes statistiques, paramétriques ou non-paramétriques, les méthodes basées sur la distance, les méthodes basées sur une métrique locale, les méthodes de clustering, les méthodes de classification, basées sur les réseaux bayésiens ou la classification à une classe, et enfin les méthodes par reconstruction.

Localisation du traitement La détection d'anomalies peut généralement être séparée en deux phases : la phase d'apprentissage, qui génère le modèle à partir de données historiques ou au fil des observations, et la phase de détection, qui applique le modèle appris pour détecter les anomalies, généralement sur de nouvelles instances.

Au sein des WSNs, ces deux phases peuvent être appliquées à différents emplacements :

— dans un noeud central, parfois même à l'extérieur du WSN,
— dans tous les noeuds du réseau de manière locale.

L'intérêt d'une approche centralisée pour l'apprentissage est de pouvoir générer un modèle global. Dans le cas de la détection, il s'agit généralement d'appliquer le modèle global sans avoir à le distribuer dans tout le réseau. Néanmoins, cela implique que tous les noeuds communiquent leurs instances vers le noeud central où est stocké le modèle.

L'apprentissage local est généralement utilisé de deux façons :

— pour générer des sous-modèles locaux transmis à un noeud central afin de construire le modèle global ; il est alors couplé à un apprentissage global,
— dans le cas d'une topologie en voisinage du réseau, où chaque modèle local peut intégrer des dépendances spatiales en communiquant avec les noeuds voisins.

En plus d'éviter de communiquer toutes les instances des noeuds dans le réseau, la détection locale permet à chaque noeud d'avoir la connaissance sur ses propres anomalies.

Construction du modèle Il existe deux approches pour construire un modèle :

— l'approche totale construit le modèle à partir de l'ensemble des instances du WSN, d'un cluster de noeud ou d'un voisinage,
— l'approche distribuée construit le modèle à partir de modèles locaux ou de métriques suffisantes définies à partir des instances ; on évite ainsi de communiquer toutes les instances à un noeud central et l'effort de calcul est également distribué dans le WSN.

Fréquence de mise à jour Pour traiter des flux de données, il est préférable que l'apprentissage ait lieu en continu et en ligne, mais différentes approches existent :

— continue : le modèle est continuellement mis à jour, à chaque nouvelle instance mesurée ; cette approche nécessite de pouvoir mettre à jour les modèles rapidement et n'est généralement pas viable avec un modèle global stocké dans un noeud central,
— régulière : les mises à jour sont réalisées avec une périodicité fixe ; cette approche est surtout utilisée dans le cas d'un apprentissage centralisé pour limiter la charge de communication qu'engendrerait une mise à jour continue ; cependant la périodicité peut ne pas être adaptée à la situation, par exemple lors d'un changement rapide de la distribution,

— adaptative : les mises à jours sont réalisées périodiquement mais la période s'adapte à la situation ; cette approche permet de résoudre le défaut d'une mise à jour régulière mais nécessite un critère bien défini pour adapter la fréquence,
— unique : dans le cas où le modèle ne peut pas être mis à jour, un réapprentissage complet du modèle est nécessaire ; la fréquence pour le réapprentissage peut suivre les trois options précédentes selon sa complexité.

Topologie du réseau Les méthodes de l'état de l'art considèrent différentes topologies pour les WSNs. Trois sont considérées, et illustrées par des figures, mais les approches pour organiser les WSNs selon ces topologies ne sont pas abordées :
— hiérarchique (Fig. 2.2) : les noeuds sont organisés selon une hiérarchie ; chaque noeud ne communique qu'avec son noeud parent et ses éventuels noeuds enfants et, en général, plus on monte dans la hiérarchie, plus les capacités des noeuds sont élevées,
— cluster (Fig. 2.3) : il s'agit d'une forme de topologie hiérarchique où les noeuds du WSN sont organisés en clusters ; chaque noeud est associé à au moins un cluster, parfois plus selon la méthode utilisée pour regrouper les noeuds entre eux, avec un noeud promu comme tête de cluster qui possède généralement des capacités de calcul plus importantes que les autres noeuds du même cluster,
— voisinage (Fig. 2.4) : aucune topologie particulière du WSN n'est assumée, mais on suppose ici que chaque noeud peut avoir connaissance de l'identité de ses noeuds voisins dans l'environnement étudié, par exemple en prenant les noeuds à portée de communication.

Certaines méthodes proposent également de s'adapter à plusieurs de ces topologies tandis que d'autres nécessitent à la fois la connaissance des noeuds voisins et une topologie hiérarchique ou en clusters.

Distribution des données dans l'environnement Dans le cas où les attributs sont les mêmes pour chaque capteur, ce qui est supposé dans cet état de l'art, la distribution des données dans l'environnement étudié par le WSN peut suivre plusieurs principes :
— homogène : la distribution est la même dans l'intégralité du réseau, ce qui signifie qu'un modèle global appris avec l'intégralité des noeuds aura la meilleure précision pour tenir compte des dépendances spatiales,
— spatiale : la distribution est la même pour des noeuds proches dans l'espace ; il est alors aussi intéressant d'apprendre des modèles globaux dans des centres de cluster, où les clusters regroupent des noeuds spatialement proches, ou à certains niveaux de hiérarchie, si les noeuds enfants d'un même parent sont proches dans l'espace, ou des modèles de voisinages locaux,
— comportementale : la distribution varie dans l'environnement, mais certains noeuds sont positionnés sur des éléments de cet environnement avec un com-

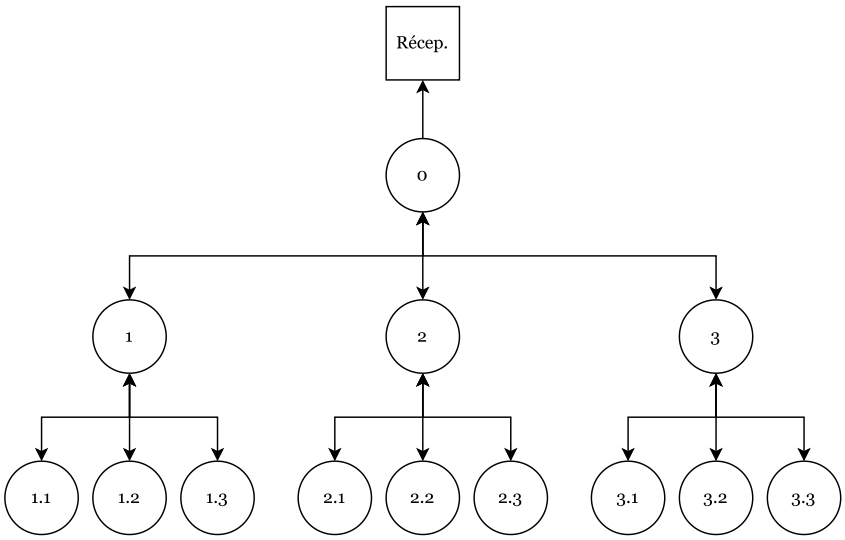

FIGURE 2.2 – Exemple de topologie hiérarchique d'un WSN avec 3 niveaux et 3 enfants par parent. Les communications n'ont lieu qu'entre parents et enfants.

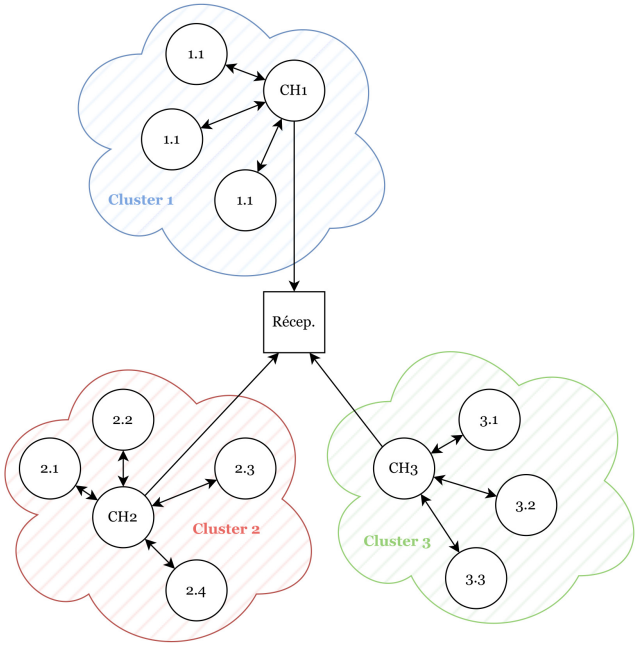

FIGURE 2.3 – Exemple de topologie en cluster d'un WSN avec 3 clusters. Les communications n'ont lieu qu'entre les têtes de clusters CH_i et les membres des clusters.

portement similaire, ce qui se traduit par des distributions similaires ; dans ce cas, une topologie en clusters, où les clusters ne seraient pas définis par la proximité des noeuds mais par la similarité des distributions, est requise pour traiter des dépendances "spatiales" (définies comme des dépendances

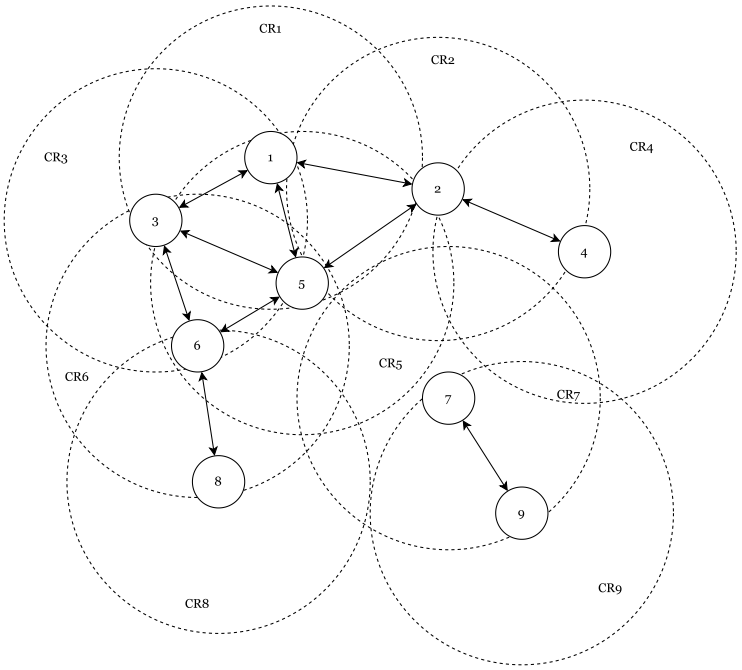

FIGURE 2.4 – Exemple de topologie en voisinage d'un WSN. Les voisinages sont définis dans un rayon de communication CR_i autour de chaque noeud.

entre différents noeuds),

— indépendante : les distributions sont toutes supposées indépendantes, et dans ce cas aucune topologie n'est nécessaire et un traitement entièrement local (apprentissage et détection) peut être utilisé sans perte d'informations sur la dépendance spatiale.

Dépendances Les différents types de dépendances étudiées par les méthodes sont intimement liés à la localisation du traitement, à la topologie supposée du réseau et à la distribution des données dans l'environnement. Chaque méthode peut étudier une ou plusieurs des trois formes de dépendances.

Les dépendances temporelles sont souvent partiellement étudiées en réalisant un apprentissage sur un ensemble d'instances générées à des instants différents. Leur étude complète est cependant plus rare et nécessite d'étudier la relation entre les instances successives.

Les dépendances spatiales sont la cible principale des méthodes de détection d'anomalies dans les WSNs et sont étudiées en construisant des modèles à partir d'un ensemble de noeuds proches.

Enfin, les dépendances entre attributs sont partiellement étudiées par les méthodes applicables aux instances multivariées. Leur étude complète est réalisée par des méthodes reposant sur les matrices de covariance par exemple.

Définition des anomalies et approximations Il est intéressant de noter comment l'anomalie est définie du point de vue de la méthode et comment elle est présentée au sein de l'article présentant cette méthode. Parfois, les articles de l'état de l'art fixent une ou plusieurs définitions et adaptent leur méthodologie pour celles-ci.

Dans le cas d'une définition implicite des anomalies, et même parfois dans le cas explicite, les méthodes peuvent réaliser des approximations dans le calcul de ces anomalies, notamment dans le cas d'approches distribuées. Deux méthodes sont donc considérées du point de vue de l'approximation ; les méthodes exactes (sans approximations) et les méthodes approchées (avec approximations).

2.2 Présentation des méthodes

Cette section présente les méthodes de détection d'anomalies appliquées aux WSNs ou aux flux de données réparties dans huit catégories : les méthodes statistiques, paramétriques et non-paramétriques, les méthodes basées sur la distance, les méthodes basées sur des métriques locales, les méthodes reposant sur un clustering, les méthodes de classification, multi-classe avec des réseaux bayésiens et à une classe avec des machines à vecteurs de support, et les méthodes par reconstruction.

2.2.1 Méthodes statistiques paramétriques

La première catégorie de méthodes traitée concerne les méthodes statistiques avec deux déclinaisons : l'approche paramétrique et l'approche non paramétrique. Ces méthodes sont, par nature, toutes approchées puisqu'on ne peut pas connaître la distribution exacte des données, seulement une distribution empirique.

Selon l'approche paramétrique, les méthodes statistiques font l'hypothèse que les données suivent une distribution pré-définie. L'objectif est alors de trouver, de manière empirique et en minimisant (ou maximisant) une métrique choisie, les paramètres de cette distribution.

2.2.1.1 Méthodes appliquées aux réseaux de capteurs

Parce que fixer une distribution pour les données peut être restrictif, certains travaux ont cherché à appliquer l'approche paramétrique non pas aux données mais à la différence entre les instances [Bettencourt 2007, Wu 2007].

Étude spatio-temporelle Dans [Bettencourt 2007], la distribution des données est supposée spatiale, avec une dépendance temporelle justifiée par une forte inertie dans les grandeurs physiques (attributs des capteurs) étudiées. Aussi, l'approche proposée cherche à tenir compte des dépendances spatiales et temporelles. Pour ce faire, les distributions de la différence entre la dernière instance d'un noeud et celles de ses noeuds voisins sont étudiées, ainsi que la distribution de la différence avec l'instance précédente. L'approche suppose donc une topologie de voisinage et chaque noeud capteur construit un modèle par distribution, soit un total de

$m + 1$ modèles dans le cas de m voisins. Ce modèle peut estimer la distribution de manière paramétrique selon plusieurs lois, dont les caractéristiques attendues sont un pic à la différence moyenne et un écrasement lorsqu'on s'en éloigne. Dans le cas d'une distribution gaussienne, suggérée dans l'article, il est possible de réaliser un apprentissage continu en mettant simplement à jour les paramètres à chaque nouvelle instance.

De plus, avec cette approche, il n'est nécessaire de stocker en mémoire, pour chaque distribution, que les paramètres du modèle et la valeur de la dernière instance. La détection d'anomalies revient alors à faire un ensemble de tests de p-valeurs et, selon les résultats agrégés des différents tests, il est possible de déduire si une anomalie est une erreur ou un évènement d'intérêt.

Puisque la méthode estime la différence entre une observation et celle de ses voisins, il est également possible de réaliser une estimation des instances manquantes en tenant compte des mesures de tous les noeuds voisins et des différences moyennes mesurées avec ces voisins.

Identification de la frontière d'un évènement De manière similaire à [Bettencourt 2007], la méthode de [Wu 2007] nécessite que chaque noeud communique avec ses voisins dans l'espace ; on se place donc également dans une topologie de voisinage. Cependant, les dépendances temporelles ne sont cette fois pas traitées et un nouveau modèle statistique est généré à chaque instant, indépendamment des modèles passés.

Pour chaque noeud, la différence de la dernière mesure et de la médiane des mesures des noeuds voisins est calculée. Le test de p-valeur est cette fois appliqué par rapport au modèle généré à partir de ces différences dans le voisinage, normalisées pour suivre une loi centrée réduite.

De plus, en réalisant un choix spécifique du voisinage, l'article propose également d'identifier les frontières des évènements d'intérêts dans un objectif de diagnostic.

2.2.1.2 Application aux flux de données

Les deux méthodes pour les WSNs présentées sont univariées en ce sens qu'elles étudient la distribution de la différence entre les instances, qui n'a de sens que pour un attribut donné. Pour tenir compte des dépendances entre attributs, des méthodes multivariées devraient être utilisées. Certaines de ces approches ont par ailleurs été appliquées dans les flux de données. Parmi elles, on trouve notamment les modèles gaussiens multivariés et, surtout, les modèles de mélanges gaussiens (GMMs) [Yamanishi 2004, Barber 2012].

Cependant, les méthodes paramétriques sont en réalité rarement applicables aux flux de données, du moins lorsqu'on cherche à modéliser les données, à cause de la nature changeante de la distribution [Thakkar 2016].

2.2.1.3 Avantages et inconvénients

Les avantages de cette sous-catégorie de méthodes sont (1) leur justification statistique, qui permet de facilement fixer un seuil de décision sur les scores d'anomalies, et (2) la légèreté des modèles, qui ne dépendent généralement que d'un faible nombre de paramètres.

Cependant, ces méthodes ont le désavantage d'être peu flexibles à la distribution réelle des données ; même lorsque les paramètres de la distribution peuvent être "appris" en continu, suivre une loi fixe est restrictif. Ne pas considérer les données elles-mêmes mais de nouveaux attributs ayant une distribution plus stable, comme la différence entre les instances, peut être une bonne alternative.

2.2.2 Méthodes statistiques non-paramétriques

Contrairement à l'approche paramétrique, l'approche non paramétrique ne suppose pas une distribution particulière et celle-ci est directement construite à partir des données, soit en utilisant des histogrammes, soit via une estimation de densité par noyau (KDE).

2.2.2.1 Application aux flux de données et fondements mathématiques

L'estimation de la distribution des données peut être réalisée, dans un cadre univarié, à travers la construction d'un histogramme. Le nombre d'éléments dans une cellule de l'histogramme est une représentation de la probabilité d'observer un point dans cette cellule. L'avantage de cette approche est que le modèle peut facilement être incrémenté avec les nouvelles instances en adaptant le nombre d'éléments dans les cellules. Pour traiter le cas multivarié, l'approche classique consiste à construire un histogramme par attribut et à calculer un score basé sur l'agrégation de ces histogrammes, à l'image de HBOS [Goldstein 2012].

Cependant, même en multivarié, l'approche par histogrammes ne permet pas de tenir compte des dépendances entre les attributs. Dans ce cas, une solution plus avancée consiste à utiliser une KDE, aussi connue comme méthode de Parzen-Rosenblatt [Parzen 1962], ajoutant une notion de continuité par rapport à l'approche par histogrammes et approximant la fonction de densité de probabilité (pdf). Soit $\mathcal{X} = \{\mathbf{x}_i, i = 1, \ldots, n\}$ un ensemble de n instances, l'estimation de la pdf f dans le cas univarié s'écrit

$$\tilde{f}_h(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{K}_h(\mathbf{x} - \mathbf{x}_i), \tag{2.1}$$

où $\mathbf{K}_h(u) = \frac{1}{h}\mathbf{K}(\frac{u}{h})$, avec $\mathbf{K}$ une fonction noyau (le noyau gaussien et le noyau de Epanechnikov étant les plus courants) et h la largeur de bande, un paramètre affectant la zone d'influence de chaque instance ou, en d'autres termes, le lissage de la courbe. Dans le cas multivarié, la formule 2.1 est étendue avec les noyaux multivariés, dont la forme générale est :

$$\mathbf{K}_\mathbf{H}(u) = |\mathbf{H}|^{-1/2}\mathbf{K}(\mathbf{H}^{-1/2}u), \tag{2.2}$$

où **H** est une matrice symétrique définie positive $p \times p$ correspondant à la largeur de bande [Langrené 2019, §2.3.1].

Les méthodes de KDEs, bien que donnant une meilleure approximation que les méthodes par histogrammes, ont tout de même une complexité qui augmente rapidement avec le nombre p d'attributs. Dans l'état de l'art, [Kristan 2011] et [Yamanishi 2004] peuvent être cités pour avoir appliqué ces méthodes aux flux de données.

2.2.2.2 Méthodes appliquées aux réseaux de capteurs

Les approches présentées dans cette section détournent l'usage des méthodes statistiques non paramétriques ; au lieu de considérer une anomalie comme une instance dans une zone de faible probabilité par rapport au modèle statistique estimé, le modèle statistique appris est utilisé pour estimer d'autres métriques définissant l'anomalie.

KDE pour estimer les (k, R)***-anomalies*** Dans le chapitre précédent, les (k, R)-anomalies de [Knorr 1998] ont été présentées au sein des méthodes basées sur la distance. Certains articles définissent des anomalies de distance selon cette approche et utilisent une KDE pour estimer à faible coût le nombre de voisins dans un rayon R autour d'une instance [Palpanas 2003, Subramaniam 2006].

L'approche de [Palpanas 2003] consiste à maintenir un modèle de KDE par noeud dans une fenêtre glissante. Le modèle de KDE est construit avec un échantillon des instances de la fenêtre comme centres de noyaux et, quand la fenêtre glisse, les noyaux peuvent changer et il faut également actualiser la largeur de bande. Une topologie hiérarchique du WSN est supposée, et la KDE dans un noeud parent utilise un échantillonnage des centres de noyaux de ses noeuds enfants et combine leurs largeurs de bandes. La méthode de [Subramaniam 2006] est similaire mais ajoute, pour limiter les communications, un paramètre limitant la fréquence à laquelle le modèle au sein d'un noeud parent est mis à jour.

La règle de Scott [Scott 1992] est utilisée pour définir la largeur de bande et le noyau de Epanechnikov est choisi car il est facile de calculer son intégrale. L'estimation du nombre de voisins dans un rayon R est en effet réalisée en intégrant la fonction issue de la KDE dans ce rayon.

Enfin, bien que le traitement soit centralisé, les deux approches utilisent des modèles locaux pour éviter d'avoir à transmettre toutes les instances des noeuds enfants vers les noeuds parents ; en effet, selon la définition de l'anomalie, une instance normale dans un noeud enfant (modèle local) ne peut être anormale dans un noeud parent (modèle central), tester les anomalies locales avec le modèle global est donc suffisant.

KDE pour estimer les anomalies de densité locale En plus des anomalies de distance, [Subramaniam 2006] définit des anomalies de densité locale comme des instances dont le facteur de déviation multi-granularité (MDEF), métrique définie

par [Papadimitriou 2003], est significativement différente de sa moyenne locale. Pour calculer cette métrique, il est nécessaire de connaître le nombre de voisins dans un rayon αr et dans tous les intervalles $2\alpha r$ du domaine, ce qui peut être estimé de la même manière que pour les anomalies de distance.

L'approche pour les anomalies de densité locale engendre cependant une complexité supérieure car, contrairement aux anomalies de distance, une instance normale pour un modèle local peut être anormale au niveau du modèle global. Pour éviter d'avoir à communiquer toutes les instances dans le noeud parent, il est donc nécessaire de transmettre le modèle global d'un parent dans chaque noeud enfant, forçant une détection uniquement locale.

Anomalies statistiques La seule approche non paramétrique recherchant des anomalies statistiques est proposée par [Bettencourt 2007] comme alternative à son approche paramétrique. En effet, au lieu d'estimer la distribution avec un modèle fixe, la solution par histogrammes est proposée. Les paramètres mis à jour sont cette fois le nombre d'éléments dans chaque cellule. Cependant, le nombre de ces paramètres est aussi bien plus élevé.

2.2.2.3 Avantages et inconvénients

En comparaison à l'approche paramétrique, l'approche non paramétrique a l'avantage d'être flexible à la distribution réelle des données.

En revanche, la légèreté des modèles n'est pas garantie ; pour l'approche des KDEs, la taille du modèle dépend du nombre de centres de noyaux considérés tandis que, pour l'approche par histogrammes, elle dépend du nombre de cellules.

2.2.3 Méthodes basées sur la distance

La seconde catégorie de méthodes concerne les méthodes basées sur la distance. Plus spécifiquement, ces méthodes s'appuient sur les (k, R)-anomalies de [Knorr 1998], mais étudient parfois également la distance au k^{th}NN [Ramaswamy 2000] ou la moyenne ou somme de celle aux kNN [Angiulli 2005].

2.2.3.1 Application aux flux de données

L'adaptation aux flux de données des méthodes basées distance revient en général à l'étude des voisinages des instances dans des fenêtres glissantes. [Tran 2016] réalise une étude comparative de six de ces méthodes : exact-Storm et approx-Storm [Angiulli 2007], Abstract-C [Yang 2009], DUE et MCOD [Kontaki 2011] et enfin Thresh_LEAP [Cao 2014]. La conclusion de cette étude est que MCOD est la plus performante de ces approches.

MCOD utilise une structure de M-tree pour indexer les instances. Cette structure facilite en outre la recherche du nombre de voisins à une certaine distance et permet un calcul exact. De plus, MCOD a la particularité d'étudier l'anormalité d'une instance durant toute sa durée de vie au sein de la fenêtre glissante et pas

seulement à l'instant de son observation. Pour réduire encore plus la complexité algorithmique, cette approche utilise une seconde structure de M-tree qui indexe cette fois des micro-clusters de tailles fixes avec un seuil sur leur nombre d'éléments, garantissant que si une instance a un cluster dans son voisinage proche, elle est normale.

De manière générale, la difficulté de ces méthodes est d'étudier les voisinages dans de grands volumes de données. L'approche de MCOD, consistant à enregistrer les mesures dans des structures facilitant la recherche des voisins, occuperait trop de mémoire dans un WSN. Pour réduire la complexité des algorithmes, beaucoup de méthodes réalisent des calculs approchés.

2.2.3.2 Méthodes appliquées aux réseaux de capteurs

Les approches appliquées aux WSNs réduisent généralement la taille des fenêtres considérées pour faciliter la recherche des voisins. Ici sont présentées à la fois des approches exactes [Sheng 2007, Branch 2006, Zhang 2007] et des approches par approximation [Palpanas 2003, Subramaniam 2006, Xie 2013].

Histogrammes pour estimer la distance au k^{th}NN De manière similaire à [Palpanas 2003] et [Subramaniam 2006], l'article de [Sheng 2007] utilise une approche statistique pour estimer les métriques nécessaires à la détection d'anomalies de distance ; seulement le modèle statistique n'est cette fois utilisé que pour modéliser les données. Deux définitions sont proposées et s'appuient sur la distance au k^{th} plus proche voisin (k^{th}NN) : la première définit une anomalie comme une instance pour laquelle cette distance est supérieure à un seuil, la seconde définit l'ensemble des anomalies comme les n instances pour lesquelles cette distance est la plus élevée (top-n anomalies).

Un histogramme est généré dans chaque noeud et tous les histogrammes sont remontés, dans une topologie hiérarchique du WSN, au noeud le plus haut qui a une capacité de traitement bien supérieure aux autres noeuds du réseau. A ce niveau, en fixant convenablement la taille des cellules de l'histogramme, il est possible de déterminer les cellules normales, les cellules anormales et les cellules potentiellement anormales. Une requête est ensuite envoyée dans le réseau pour que chaque noeud fasse remonter les instances dans les cellules anormales et potentiellement anormales.

Confirmer l'anormalité des instances dans les cellules potentiellement anormales étant coûteux, il est proposé de réitérer les requêtes d'histogrammes en faisant varier la taille des cellules pour réduire le nombre de cellules potentiellement anormales. L'approche reste cependant lourde, applicable uniquement attribut par attribut et un ré-apprentissage complet est nécessaire à chaque requête. Cette méthode permet néanmoins la recherche exacte des anomalies telles que définies sans avoir à calculer exactement la distance au k^{th}NN.

Calcul exact de la distance au k^{th}NN Contrairement à [Sheng 2007] qui trouve les anomalies basées sur la distance au k^{th}NN sans avoir à la calculer exactement, les approches de [Branch 2006, Zhang 2007] réalisent un calcul exact de cette distance.

Plus largement, [Branch 2006], complété par [Branch 2013], permet de s'adapter à trois définitions différentes de l'anomalie qui se basent respectivement sur :

— la distance au k^{th}NN,
— la distance moyenne aux kNN,
— l'inverse du nombre de voisins à une distance R.

De son côté, l'article de [Zhang 2007] ne considère que la distance au k^{th}NN et définit plus précisément les anomalies comme les top-n anomalies dans une fenêtre glissante, ce qui évite de fixer un seuil sur la distance.

Ces approches définissent un *rang de déviation* comme un classement sur le score d'anomalies avec des propriétés particulières, et qui peut être utilisé pour les différentes anomalies définies. En outre, lorsqu'un ensemble est contenu dans un autre, le rang de déviation du plus grand ensemble sera toujours inférieur ou égal à celui du plus petit. Cette propriété permet d'itérer sur la détection des anomalies en affinant les résultats à chaque itération jusqu'à convergence. Entre chaque itération, ce sont les anomalies théoriques qui sont transmises dans le réseau. Afin de réduire le coût de communication, il est également nécessaire de définir des ensembles supports comme un ensemble d'instances nécessaires au maintien du rang de déviation. Ce sont précisément ces ensembles qui sont transmis.

Dans [Branch 2006], la topologie du WSN est celle en voisinages. Chaque noeud transmet ses observations à ses voisins, et donc tous les noeuds ont en mémoire leurs observations et celles de leurs voisins. Les anomalies initiales sont calculées dans chaque noeud et sont transmises, avec l'ensemble support associé, aux voisins. Si un noeud découvre de nouvelles anomalies parmi celles transmises par un voisin, il met à jour son modèle et le transmet à ses autres voisins. Suite aux propagations successives de ces informations, chaque noeud finit éventuellement par disposer de la connaissance des anomalies globales. Pour mettre à jour le modèle, il suffit que toutes les instances transmises soient accompagnées de leur date de mesure ; chaque noeud supprime alors les plus anciennes, met à jour son modèle et le transmet à ses voisins. On peut considérer qu'il s'agit d'une stratégie entièrement locale où tous les noeuds du réseau sont l'équivalent de noeuds centraux d'une stratégie centralisée suite à la transmission de fragments du modèle entre voisins.

L'approche de [Zhang 2007] est similaire mais en utilisant une topologie hiérarchique. L'itération se fait entre la racine et les feuilles de l'arbre associé à cette hiérarchie. L'avantage par rapport à l'approche de [Branch 2006] est qu'on sait exactement quand la phase itérative a convergé. Cette méthode s'appuie également sur des fenêtres glissantes et permet la mise à jour continue du modèle, mais propose également une version avec un temps d'attente de taille fixe entre les mises à jour pour limiter la charge de communications.

Selon la taille des fenêtres et la valeur du paramètre k, les approches exactes peuvent être très lourdes en calculs et en communications.

Approximation statistique du nombre de voisins Deux approches pour détecter des anomalies de distance en s'appuyant sur des méthodes statistiques ont déjà été présentées dans les sections précédentes [Palpanas 2003, Subramaniam 2006]. L'anomalie y était définie par rapport au nombre de voisins dans un rayon R.

Approximation du nombre de voisins avec des hyper-cubes Une autre approche pour dénombrer les voisins dans un certain périmètre, et plus précisément dans un hyper-cube de diagonale d, a été proposée par [Xie 2013]. Pour utiliser cette approche, les données doivent pouvoir être normalisées en temps réel. L'article définit des hyper-grilles qui permettent de détecter les anomalies très facilement avec des opérations binaires. Ainsi, chaque observation tombe dans une cellule (un hyper-cube) de diagonale $d/2$ qui connaît le nombre d'éléments qu'elle contient. En étudiant les éléments d'une cellule et des cellules autour d'elles, il est possible d'assurer que certaines cellules ne contiennent que des éléments normaux ou anormaux. Pour les cellules restantes, le nombre de voisins est approximé en utilisant des hyper-cubes de substitution.

L'approche est appliquée à une topologie en cluster ; le dénombrement dans l'hyper-grille est d'abord réalisé dans chaque noeud puis transmis à la tête de cluster qui génère le modèle global et le redistribue dans les noeuds. La mise à jour du modèle global ne se fait qu'avec une certaine probabilité.

2.2.3.3 Avantages et inconvénients

Le principal inconvénient de ces méthodes est la difficulté à étudier les voisinages des instances, en particulier quand le nombre de dimensions augmente. Cependant, cette problématique peut être allégée par l'utilisation de méthodes approchées ou par la réduction de la taille des fenêtres d'instances considérées.

Les approches basées sur la distance entre les instances sont en revanche appréciées car les anomalies de distance sont très facilement interprétables et offrent des propriétés intéressantes pour les approches distribuées ; entre autre, lorsque deux ensembles sont fusionnés, les anomalies des deux ensembles peuvent devenir normales mais les instances normales ne peuvent pas devenir anormales [Branch 2006, Zhang 2007].

2.2.4 Méthodes basées sur des métriques locales

De façon similaire aux méthodes basées sur la distance, les méthodes basées densité locale étudient le voisinage des instances. Cependant, plutôt que de s'appuyer sur la distance d'une instance à ses voisins, elles calculent des métriques mesurant la répartition des instances dans le voisinage.

2.2.4.1 Fondements mathématiques

La métrique de densité locale la plus populaire est le Local Outlier Factor (LOF) défini par [Breunig 2000]. Son calcul pour une instance est dépendant de son k^{th}NN et de ses kNN. Soit $\mathbf{x}$ une instance, $D_k(\mathbf{x})$ sa distance à son k^{th}NN, $\mathcal{N}_k(\mathbf{x})$ ses kNN et $d(\mathbf{x}, \mathbf{y})$ la distance entre $\mathbf{x}$ et une seconde instance $\mathbf{y}$. La distance d'accessibilité de $\mathbf{x}$ par rapport à une instance $\mathbf{y} \in \mathcal{N}_k(\mathbf{x})$ est définie par

$$rd_k(\mathbf{x}, \mathbf{y}) = \max(d(\mathbf{x}, \mathbf{y}), D_k(\mathbf{y})) \tag{2.3}$$

et la densité d'accessibilité locale de $\mathbf{x}$ est définie comme l'inverse de sa distance d'accessibilité moyenne à ses kNN :

$$lrd_k(\mathbf{x}) = \frac{1}{\sum_{\mathbf{y} \in \mathcal{N}_k(\mathbf{x})} rd_k(\mathbf{x}, \mathbf{y})/k} \,. \tag{2.4}$$

Enfin, le LOF de $\mathbf{x}$ pour un paramètre k fixé est défini comme le rapport de la moyenne des densités d'accessibilité locales de ses kNN sur sa densité d'accessibilité locale

$$\text{LOF}_k = \frac{\frac{1}{k} \sum_{\mathbf{y} \in \mathcal{N}_k(\mathbf{x})} lrd_k(\mathbf{y})}{lrd_k(\mathbf{x})} \,. \tag{2.5}$$

Le calcul du LOF d'une instance nécessite donc d'avoir calculé sa densité d'accessibilité locale ainsi que celle de ses voisins, ce qui nécessite donc le calcul des distances d'accessibilité de l'instance par rapport à ses voisins et de ses voisins par rapport à leurs propres voisins. Il est ainsi nécessaire de calculer la distance au k^{th}NN pour les voisins des voisins de l'instance dont on souhaite calculer le LOF, ce qui force finalement à conserver en mémoire ces métriques pour chaque instance du jeu de données. Aussi, si le LOF a grandement été utilisé dans le cas de données statiques, son application au cas dynamique des flux de données a aussi motivé de nombreux travaux du fait de sa complexité.

2.2.4.2 Application aux flux de données

En démontrant que l'ajout ou la suppression d'une instance dans un jeu de données n'affectait que les métriques associées aux instances dans un voisinage restreint, une première implémentation d'un LOF incrémental (iLOF) a vu le jour [Pokrajac 2007]. Cette approche propose un calcul exact du LOF en mettant à jour les métriques de toutes les instances touchées par l'ajout ou la suppression de l'instance. Il est ainsi possible, par exemple, de calculer le LOF des nouvelles instances dans une fenêtre glissante pour un flux de données. Cependant, la complexité algorithmique reste élevée, et les instances sont stockées dans un M*-tree pour faciliter la recherche des kNN avec toutes leurs métriques associées gardées en mémoire.

Plus tard, le LOF incrémental amélioré (I-IncLOF) est proposé par [Karimian 2012] pour réduire le taux de faux positifs de iLOF, causé par la suppression des comportements anciens, tout en réduisant le temps de traitement ; au lieu de prendre la décision d'anomalie lorsqu'une instance est insérée dans le jeu de

données, la décision est prise dans une fenêtre glissante (de taille très inférieure à la taille du jeu de données), comme pour MCOD, présenté en Section 2.2.3.1. Si le score d'anormalité se stabilise, c'est qu'il s'agissait d'un nouveau comportement normal. Alors que iLOF proposait de supprimer les instances anciennes (sortant de la fenêtre glissante modélisant le jeu de données), celles-ci ne sont pas toutes supprimées dans I-IncLOF ; seules les anomalies sont supprimées. Cependant, le désavantage de cette approche est qu'elle introduit un délai dans la détection et que l'occupation mémoire augmente au cours du temps.

Tout comme I-IncLOF, les approches qui ont suivi ont eu pour objectif d'améliorer iLOF sans perdre la connaissance des instances anciennes, mais cette fois en fixant l'occupation mémoire [Salehi 2016, Na 2018, Huang 2020]. La solution envisagée par ces nouvelles approches consiste à résumer une partie du jeu de données avec un nombre limité d'instances à chaque fois que l'occupation mémoire atteint un certain seuil. MiLOF [Salehi 2016] propose de résumer une partie des instances sous la forme de centres de clusters, mais perd ce faisant la densité associée aux clusters. DILOF, proposé par [Na 2018], sélectionne intelligemment les instances à supprimer, parmi les plus anciennes, pour minimiser la différence de densité entre l'ensemble initial et l'ensemble final. Enfin, TADILOF [Huang 2020] propose une amélioration de DILOF où le problème d'optimisation, qui permet de sélectionner les instances à conserver, tient également compte de l'ancienneté des instances pour permettre de considérer les changements de distribution.

2.2.4.3 Méthodes appliquées aux réseaux de capteurs

Une approche pour calculer une métrique de densité locale dans le cas des WSNs en s'appuyant sur une méthode statistique a déjà été présentée [Subramaniam 2006]. La métrique utilisée, le facteur de déviation MDEF, a été proposée dans [Papadimitriou 2003] au sein de la méthode LOCI, dont l'objectif est de détecter des anomalies dans des jeux de données multivariés et de grandes tailles. Le MDEF est plus intuitif et plus simple à calculer que le LOF, et s'appuie sur le voisinage dans un rayon R plutôt que sur le voisinage défini par les kNN. En théorie, il est nécessaire de dénombrer les instances dans le voisinage d'une instance et dans le voisinage de ses instances voisines, mais ces dénombrements sont estimés ici à partir de la densité de la distribution, elle-même estimée par KDE.

2.2.4.4 Avantages et inconvénients

Tout comme pour les méthodes basées sur la distance, celles reposant sur la densité locale ont l'inconvénient de nécessiter l'étude de voisinages. Bien qu'elles soient souvent moins interprétables, leur principal avantage par rapport aux méthodes basées sur la distance est leur robustesse face à des distributions avec plusieurs densités.

2.2.5 Méthodes reposant sur un clustering

Les méthodes de clustering forment la quatrième catégorie étudiée. Le problème du clustering dans les flux de données a été traité dans de nombreux articles, dont certains seront mentionnés dans cette section, mais assez rarement avec une ambition de détection d'anomalies.

2.2.5.1 Application aux flux de données

Le clustering dans un jeu de données construit incrémentalement, comme les flux de données, est appelé clustering dynamique. Il existe de nombreuses méthodes de clustering dynamique, et les études [Zhang 2013a] et [Salehi 2018] réalisent une description plus avancée de cette catégorie d'approches.

L'applicabilité de ces approches à la détection d'anomalies est nuancée par le fait qu'il ne s'agit pas de leur objectif premier [Thakkar 2016]. En effet, comme mentionné dans le chapitre précédent, le passage du clustering à la détection d'anomalies repose sur certaines hypothèses. En rendant le clustering dynamique, les méthodes utilisent des approches permettant l'évolution des clusters tout en limitant la quantité d'information en mémoire pour les définir, et les hypothèses pour le passage à la détection d'anomalies peuvent ne plus être viables.

Quelques approches utilisées pour le clustering dynamique peuvent tout de même être brièvement présentées ici :

— BIRCH [Zhang 1996] et CluStream [Aggarwal 2003] utilisent des caractéristiques suffisantes des clusters pour pouvoir les estimer sans stocker leurs éléments ; ces caractéristiques contiennent notamment le nombre d'éléments dans chaque cluster ainsi que la somme et la somme des carrés des valeurs,
— DenStream [Cao 2006] et SDstream [Ren 2009] reprennent le principe de CluStream mais, au lieu de réaliser un clustering selon la distance, qui produit des clusters circulaires, ces approches utilisent la densité pour former des clusters de formes arbitraires ; aussi, les deux méthodes utilisent l'approche DBSCAN [Ester 1996] pour générer les clusters finaux,
— D-Stream [Chen 2007] est similaire à DenStream et SDstream mais utilise des grilles de densité,
— DyClee [Roa 2019] réalise également un premier clustering selon la distance pour obtenir des micro-clusters et un second selon la densité pour obtenir des clusters finaux de formes arbitraires ; les clusters ainsi produits peuvent également se déplacer en suivant les déviations de comportement.

2.2.5.2 Méthodes appliquées aux réseaux de capteurs

Les méthodes décrites dans cette sous-section présentent un apprentissage unique du clustering ; il ne s'agit donc pas d'approches de clustering dynamique. La construction est toutefois distribuée, et l'objectif est d'obtenir un clustering global pour le WSN.

Clustering distribué utilisant des hyper-sphères Deux approches sont proposées pour le clustering distribué dans un WSN à partir d'hyper-sphères [Rajasegarar 2006, Rajasegarar 2014].

Dans [Rajasegarar 2006], un clustering en hyper-sphères de rayons fixes est réalisé dans chaque noeud. Des caractéristiques suffisantes des clusters sont ensuite transmises aux noeuds parents qui fusionnent les clusters identifiés par leurs enfants jusqu'à remonter au noeud le plus haut de la hiérarchie. A ce niveau, le modèle de clustering global peut être estimé.

Cependant, l'article propose une normalisation des données, et la fusion des clusters provenant de différents noeuds suppose que la normalisation soit la même dans tous les noeuds, ce qui implique que le noeud le plus élevé de la hiérarchie calcule les paramètres de normalisation et les transmette dans le réseau.

La fusion des clusters nécessite aussi de fixer une distance et un seuil entre les clusters proches. Ceux-ci étant de tailles fixes, l'approche proposée est de vérifier que la distance entre les centres des hyper-sphères est inférieure au rayon ; pour fusionner deux clusters, on crée un troisième cluster dont le centre est le point milieu des deux précédents centres. Cette approche, bien que légère, est cependant approximative.

Enfin, pour détecter les anomalies dans le modèle global, la distance moyenne de chaque cluster $\mathbf{c}$ à ses kNN clusters $\mathcal{N}_k(\mathbf{c})$, notée $d(\mathbf{c}, \mathcal{N}_k(\mathbf{c})$, est étudiée. Soit $\mu_{C,k}$ la moyenne des $d(\mathbf{c}, \mathcal{N}_k(\mathbf{c}))$ pour l'ensemble C des clusters et $\sigma_{C,k}$ son écart-type, un cluster $\mathbf{c}$ est considéré anormal si $d(\mathbf{c}, \mathcal{N}_k(\mathbf{c})) > \mu_{C,k} + \sigma_{C,k}$. Les clusters anormaux sont ensuite transmis aux noeuds pour qu'ils puissent réaliser la détection des anomalies avec les instances en mémoire.

Le second article [Rajasegarar 2014] complète le précédent avec les modifications suivantes :

— une détection d'anomalie locale est également réalisée au niveau des noeuds avec le clustering local,
— le seuil sur la distance pour fusionner les clusters peut désormais être inférieur au rayon des hypersphères, limitant ainsi l'approximation mais augmentant le nombre de clusters dans le noeud racine de la structure,
— pour la détection d'anomalies, un paramètre ψ est ajouté et on considère une instance anormale si $d(\mathbf{c}, \mathcal{N}_k(\mathbf{c})) > \mu_{C,k} + \psi \times \sigma_{C,k}$.

Clustering distribué utilisant des hyper-ellipsoïdes Une approche plus flexible consiste à utiliser des hyper-ellipsoïdes, dont les hyper-sphères sont des cas particuliers, pour le clustering [Moshtaghi 2009]. La motivation derrière cette approche est que l'utilisation d'hyper-ellipsoïdes, qui s'accompagne de l'utilisation de la distance de Mahalanobis et de la matrice de covariance de la distribution, permet de tenir compte des dépendances entre attributs.

L'approche ne suppose pas cette fois l'homogénéité des distributions des différents noeuds ; chaque noeud modélise ses données sous la forme d'un unique cluster ellipsoïdal, et en déduit également les anomalies locales comme les instances à l'ex-

térieur de l'hyper-ellipsoïde. Des caractéristiques suffisantes des hyper-ellipsoïdes générées dans l'ensemble des noeuds sont remontées jusqu'au noeud le plus élevé d'une topologie hiérarchique. Ce noeud est chargé de réaliser un clustering global des hyper-ellipsoïdes ; les hyper-ellipsoïdes fusionnées sont donc liées à des noeuds ayant un comportement similaire, la distribution dans l'environnement est ainsi supposée comportementale.

Le clustering global nécessite de fixer deux paramètres :

— une mesure de distance entre deux hyper-ellipsoïdes ; la mesure utilisée dans l'article est basée sur les centres des hyper-ellipsoïdes,
— le nombre de clusters ; déterminé de manière optimale par un diagramme PRE (pour Positive Root Eigenvalue).

Un clustering hiérarchique est ensuite utilisé et l'article propose une approche pour fusionner les hyper-ellipsoïdes appartenant au même cluster, permettant ainsi d'obtenir le modèle global qui est redistribué dans les noeuds afin d'y détecter les anomalies globales.

Clustering utilisant DBSCAN La dernière approche présentée dans cette section utilise la méthode DBSCAN, proposée dans [Ester 1996], avec quelques modifications et pour la problématique des WSNs [Abid 2017]. L'approche est entièrement centralisée et, à la fin du clustering, la source des instances dans chaque cluster est étudiée.

Cette méthode supposant l'homogénéité des données dans l'environnement, si un cluster ne contient que des instances provenant du même noeud capteur, alors le cluster entier est considéré comme anormal. Les instances considérées comme du bruit par l'algorithme de clustering sont également classées comme anormales.

2.2.5.3 Avantages et inconvénients

L'utilisation des approches de clustering pour la détection d'anomalies repose sur certaines hypothèses dans le cas hors ligne. En revanche, bien que des méthodes de clustering dynamique aient été développées, leur utilisation pour la détection d'anomalies dans les flux de données est critiquée. Ainsi, dans le cas des WSNs, les méthodes existantes sont statiques et nécessitent au mieux un ré-apprentissage régulier.

2.2.6 Méthodes de classification avec des réseaux bayésiens

Les méthodes de classification ne sont généralement pas applicables au cadre non supervisé car leur apprentissage nécessite d'utiliser les labels des classes. Il existe cependant deux exceptions parmi ces approches : les réseaux bayésiens pour la classification multi-classes et les machines à vecteurs de support pour la classification à une classe.

Les réseaux bayésiens mentionnés ici ont la particularité de définir les classes en découpant l'espace des données. Ainsi, chaque instance est naturellement labellisée

pour ces classifieurs. L'objectif est alors de déterminer la probabilité postérieure (ou a posteriori) qu'une instance tombe dans une classe à partir de la connaissance de l'état du réseau à cet instant.

Les réseaux bayésiens prennent la forme de graphes orientés acycliques où chaque noeud correspond à une variable aléatoire. Leur construction nécessite de générer le graphe des relations entre les variables et des tableaux de probabilités associés.

Il existe trois types d'approches de réseaux bayésiens dans l'état de l'art : les classifieurs bayésiens naïfs, définis comme une simplification des réseaux bayésiens, les réseaux bayésiens dynamiques, qui intègrent une composante permettant d'étudier l'évolution des variables du réseau bayésien dans le temps, et les diagrammes causaux, où les liens entre les noeuds sont représentés par des relations de causalité.

2.2.6.1 Méthodes pour les réseaux de capteurs

Les trois types de réseaux bayésiens mentionnés ont été appliqués pour la détection d'anomalies dans les WSNs. Cependant, seuls les classifieurs bayésiens naïfs et les diagrammes causaux ont été employés sans apprentissage supervisé.

Classifieurs bayésiens naïfs Il existe plusieurs approches dans l'état de l'art ayant utilisé les classifieurs bayésiens naïfs (CBNs), facilement applicables pour leur simplicité [Elnahrawy 2004, Titouna 2015, Titouna 2019]. Dans un CBN, un noeud du graphe correspond à la variable cible C et les autres noeuds à n variables caractéristiques $X_1, ..., X_n$, comme représenté dans la Figure 2.5. L'espace des mesures pour chaque variable est discrétisé pour prendre la forme d'un nombre fini de classes. L'objectif du classifieur est d'inférer la probabilité conditionnelle $P(C|X_1, ..., X_n)$. Avec les hypothèses des CBNs, cela revient à inférer $P(C)$ et les $P(X_i|C)$ pour $1 \leq i \leq n$.

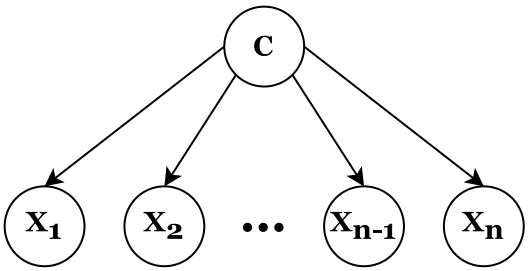

FIGURE 2.5 – Représentation du graphe d'un classifieur bayésien naïf avec une variable cible C et n variables caractéristiques $X_1, X_2, ..., X_{n-1}, X_n$

L'approche de [Elnahrawy 2004] vise à assurer la qualité et la fiabilité des mesures dans un WSN, en remplaçant notamment les valeurs manquantes. La détection d'anomalies n'est ainsi pas un objectif principal mais est possible grâce aux inférences que permettent les CBNs.

La variable cible correspond à l'instance mesurée par le noeud à l'instant actuel et deux variables caractéristiques sont considérées : une variable historique H

correspondant à l'instance précédente pour tenir compte des dépendances temporelles, une variable N correspondant aux instances des noeuds voisins pour traiter les dépendances spatiales.

L'inférence des probabilités est faite par dénombrement pour obtenir la fréquence d'occurrence des différentes classes, ce qui peut être réalisé sous la forme d'un histogramme. Pour $P(C)$, il faut stocker le nombre d'occurrences des classes $\{c_i\}$. S'il y a m classes différentes pour C alors, pour $P(H|C)$, il faut stocker jusqu'à $m \times m$ nombre d'occurrences (chaque combinaison).

En supposant qu'il y ait également m classes possibles pour les instances de chaque voisin, alors le nombre d'occurrences à stocker pour estimer $P(N|C)$ est m^{l+1} s'il y a l voisins. Pour réduire le nombre de compteurs à stocker, [Elnahrawy 2004] mentionne un modèle de deux voisins indistinguables, ce qui réduit le nombre de compteurs à $\frac{m^2(m+1)}{2}$.

Les modèles sont ainsi lourds en mémoire. Cependant, ils ont l'avantage de pouvoir être appris en continu en incrémentant le nombre d'occurrences, à l'image des histogrammes.

Deux topologies sont considérées :

— en voisinage dans le cas d'une distribution spatiale ; les inférences sont réalisées dans chaque noeud sans communication dans le réseau en dehors des valeurs communiquées entre noeuds voisins,

— hiérarchique dans le cas d'une distribution homogène ; les enfants peuvent partager leurs compteurs (ou histogrammes) avec leurs parents pour générer un modèle global, cependant le coût en communication est très élevé au vu du nombre de paramètres à transmettre, et la mise à jour du modèle global ne peut donc se faire que périodiquement.

La classe maximisant la probabilité a posteriori $P(C|H, N)$ peut être utilisée pour remplacer une valeur manquante, et la détection d'anomalies étudie le rapport entre la probabilité postérieure de la classe observée et celle maximisant la probabilité a posteriori.

L'article de [Titouna 2015] propose une approche de détection d'anomalies en deux phases : une détection locale et une détection globale.

La détection locale est similaire à l'approche par voisinage de [Elnahrawy 2004] mais en se limitant à un unique voisin et en ajoutant une troisième variable caractéristique correspondant à l'instance précédente du voisin. De plus, la détection d'anomalies locales ne regarde pas le rapport des probabilités mais seulement si l'instance observée appartient à la classe maximisant la probabilité a posteriori.

L'approche globale, quant à elle, suppose une topologie en cluster et a lieu dans la tête de cluster. Cette approche nécessite que tous les noeuds transmettent leurs mesures et la décision locale à la tête de cluster. Les anomalies globales sont déterminées en tenant compte de la distance entre les mesures des différents noeuds à un instant donné et la décision locale, permettant possiblement de considérer les évènements d'intérêt.

Enfin, dans [Titouna 2019], la structure du graphe ne considère pas cette fois les dépendances spatiales mais seulement la dépendance temporelle avec l'instance pré-

cédente. Cependant, les dépendances spatiales sont tout de même prises en compte par l'approche en supposant l'homogénéité de la distribution dans l'espace et en réalisant un apprentissage hors ligne sur le réseau. Deux variables caractéristiques sont considérées : la première correspond aux mesures historiques et la seconde à l'état de charge de la batterie du capteur, qui peut influencer les mesures. De plus, un classifieur est maintenu pour chaque variable, ce qui empêche de tenir compte des dépendances entre les attributs.

Pour les approches de [Elnahrawy 2004] et [Titouna 2015], un traitement multivarié, en tenant donc compte des dépendances entre attributs, est possible mais augmenterait rapidement le nombre de paramètres, et donc l'occupation mémoire, en augmentant le nombre de classes.

Aussi, dans [Titouna 2015] et [Titouna 2019], les instances normales sont remontées jusqu'à la station de base tandis que les anomalies sont rejetées.

Diagrammes causaux Contrairement aux CBNs, l'utilisation des diagrammes causaux (DCs) permet de tenir compte des dépendances entre les attributs. Le réseau bayésien est construit de manière similaire à celui présenté en Figure 2.5, mais les différentes probabilités à inférer ($P(C)$ et les $P(X_i|C)$) dépendent des autres attributs. Dans le DC qui définit cette dépendance entre attributs, chaque noeud du graphe correspond à un attribut et les arcs représentent des relations causales.

Supposons qu'on ait cinq attributs A, B, C, D, E, la probabilité jointe, correspondant à celle de l'instance, peut s'écrire $P(A, B, C, D, E) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)P(E|A, B, C, D)$. Cependant, la dépendance entre les attributs, représentée dans un DC, peut simplifier le calcul de la probabilité jointe. La Figure 2.6 représente un tel diagramme, la probabilité jointe peut y être reformulée comme $P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|A, C)P(E|A, C)$ car C ne dépend que de A conditionnellement et pas de B, etc.

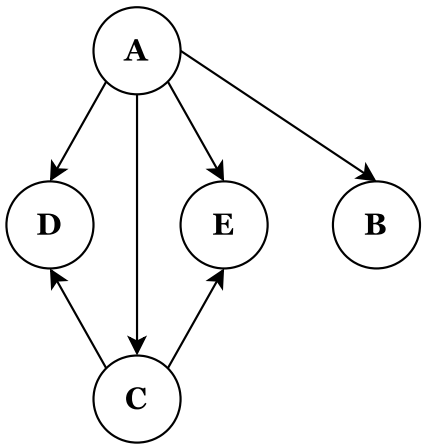

FIGURE 2.6 – Représentation d'un diagramme causal pour cinq attributs A, B, C, D, E

L'article de [Janakiram 2006] décrit l'utilisation des DCs dans la détection d'anomalies pour les WSNs. Le réseau bayésien utilisé prend la même forme que celui de [Elnahrawy 2004] mais en ajoutant la relation de la variable cible pour un attribut aux mesures des autres attributs. Ainsi, l'anomalie peut être détectée sur chaque attribut en tenant compte des autres.

L'approche nécessite deux phases importantes :

— la construction du DC ; simple lorsque la structure est fournie car il ne reste qu'à estimer les probabilités conditionnelles, mais rapidement difficile sans connaissances sur les relations de dépendances entre attributs,

— la construction des tables de probabilité ; dans le cas des diagrammes causaux, le nombre de tables de probabilités augmente et, pour chaque attribut, il en faut (1) pour les dépendances avec les autres attributs, (2) pour la dépendance avec les mesures historiques et (3) pour la dépendance avec les mesures des voisins.

Pour simplifier le modèle, on considère un nombre fixe de voisins. Bien que l'approche permette de considérer les dépendances entre attributs, le nombre de paramètres en mémoire est bien plus élevé que pour l'approche de [Elnahrawy 2004] et croit rapidement avec le nombre d'attributs.

2.2.6.2 Avantages et inconvénients

Le principal avantage des méthodes de classification utilisant des réseaux bayésiens est leur formalisation mathématique des différentes formes de dépendances.

Cependant, les modèles sont souvent très lourds en mémoire puisqu'ils nécessitent de stocker un nombre important de compteurs pour estimer les probabilités conditionnelles de chaque classe. Ce nombre augmente rapidement si on considère plusieurs attributs dans le cas des CBNs, c'est pourquoi on se contente d'un modèle par attribut. Les DCs permettent d'intégrer directement les dépendances entre attributs, mais il est nécessaire de déterminer les dépendances conditionnelles entre attributs pour limiter le nombre de paramètres en mémoire, ce qui peut être difficile sans connaissances métiers et suppose une heuristique particulière.

2.2.7 Méthodes de classification à une classe

Les approches de classification à une classe présentées dans cette section s'appuient sur les machines à vecteurs de support (OCSVM pour One-Class SVM).

Les machines à vecteurs de support multi-classes sont supervisées et apprennent à séparer les différentes classes connues à partir de données d'apprentissages labellisées, mais les OCSVM définissent un contour autour des données d'apprentissage seulement. Cette approche est particulièrement intéressante dans le domaine de la détection de nouveautés où on cherche à détecter les nouvelles instances s'écartant des données d'apprentissage.

Pour la détection d'anomalies, le risque est d'intégrer au modèle appris des anomalies présentes dans les données d'apprentissage ; pour palier ce problème, la

plupart des méthodes OCSVM définissent un paramètre de liberté permettant de laisser des instances à l'extérieur du contour appris.

2.2.7.1 Fondements mathématiques

Les méthodes étudiées dans cette section utilisent des OCSVM sphériques (reposant sur la distance euclidienne) ou ellipsoïdaux (reposant sur la distance de Mahalanobis). L'objectif de ces approches est d'apprendre un contour non linéaire autour des données à partir d'un contour sphérique ou ellipsoïdal dans un espace de plus grande dimension appelé *espace de représentation*.

Soit $\phi(\mathbf{x})$ l'image d'une instance $\mathbf{x}$ dans l'espace de représentation, $d(.,.)$ la distance dans l'espace de représentation et c le centre de l'hyper-sphère ou de l'hyper-ellipsoïde, trouver les paramètres de la séparation revient à résoudre le problème :

$$\begin{aligned} \text{minimiser :} \quad & R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \\ \text{avec :} \quad & d(\phi(\mathbf{x}_i), c) \leq R^2 + \xi_i \\ & \xi_i \geq 0, \quad i = 1, ..., n \end{aligned} \tag{2.6}$$

où les ξ_i sont des variables de relaxation introduites pour permettre à quelques instances (anormales) de se trouver à l'extérieur du contour et dont le nombre est régulé par le paramètre ν. Enfin, R correspond au rayon de l'hyper-sphère ou au rayon effectif de l'hyper-ellipsoïde.

En particulier, dans le cas d'une hyper-ellipsoïde, on utilise la distance de Mahalanobis à la distribution des instances ; on prend alors $c = \mu = \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)$ et $d(\phi(\mathbf{x}_i), c) = (\phi(\mathbf{x}_i) - \mu)\Sigma^{-1}(\phi(\mathbf{x}_i) - \mu)^T$ avec Σ la matrice de covariance de la distribution.

Dans le cas d'une hyper-sphère, cela revient à utiliser la matrice identité à la place de la matrice de covariance, et on obtient la norme euclidienne $d(\phi(\mathbf{x}_i), c) = \|\phi(\mathbf{x}_i) - c\|^2$.

Pour résoudre ces problèmes dans l'espace de représentation, l'astuce du noyau est utilisée pour éviter de réaliser des calculs en grande dimension. En outre, il n'est même pas nécessaire de connaître la fonction ϕ. A la place, on utilise un noyau K vérifiant $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})\phi(\mathbf{y})^T$.

L'approche reposant sur les hyper-sphères est la plus répandue en détection d'anomalies et connue sous le nom de SVDD (pour Support Vector Data Description) [Tax 2004].

2.2.7.2 Méthodes pour les réseaux de capteurs

Le problème décrit précédemment est généralement quadratique. Cependant, en centrant l'hyper-sphère ou l'hyper-ellipsoïde, il est possible de se ramener à un problème linéaire, plus facile à résoudre avec une faible puissance de calcul, comme proposé par [Laskov 2004] avec des quarts de sphères (les instances dans l'hyper-sphère centrée se retrouvant dans une même partie de l'espace).

Représentation en quarts de sphères La majorité des travaux pour la classification à une classe dans les WSNs utilise les OCSVM à quarts de sphères ou QSSVM [Rajasegarar 2007, Zhang 2008, Zhang 2009, Shahid 2012a, Shahid 2012b, Yao 2015].

En supposant une topologie hiérarchique et une distribution homogène dans l'environnement, [Rajasegarar 2007] propose une première approche où chaque noeud réalise l'apprentissage d'un modèle de QSSVM en résolvant le problème linéaire avec ses instances. Les noeuds conservent alors en mémoire les normes de chaque instance dans l'espace de représentation ainsi que le rayon de l'hyper-sphère apprise. Pour les anomalies locales, on a $\|\phi(\mathbf{x}_i)\|^2 = K(\mathbf{x}_i, \mathbf{x}_i) > R^2$.

Les rayons calculés dans les différents noeuds enfants sont remontés aux noeuds parents qui génèrent un modèle global en agrégeant les rayons. Plusieurs approches sont proposées pour l'agrégation : le minimum, le maximum, la médiane ou la moyenne. Le rayon du modèle global est redistribué dans le réseau et permet d'estimer les anomalies globales dans chaque noeud.

L'approche proposée n'est cependant pas en ligne ; pour évaluer de nouvelles instances, il faut relancer le processus dans une nouvelle fenêtre. Elle peut en revanche être appliquée à d'autres topologies tant qu'un noeud a la responsabilité d'un ensemble d'autres noeuds.

Une approche similaire est proposée par [Zhang 2008] avec une topologie de voisinage cette fois et en considérant deux modèles globaux pour différencier les erreurs des évènements. Pour ce faire, en plus du premier voisinage de communication, un second voisinage contenant les voisins les plus proches est défini. La détection d'anomalies est toutefois similaire mais implique plus de communications puisque chaque noeud agrège les rayons de ses noeuds voisins. La méthode d'agrégation retenue est la médiane. L'article ajoute également une méthode de calcul de la norme de nouvelles instances, ce qui permet de réaliser une détection en ligne.

Dans [Zhang 2009], trois approches sont proposées pour l'apprentissage en ligne des modèles :

- la première approche consiste à ré-apprendre le modèle à chaque intervalle de temps, en conservant les instances en mémoire dans une fenêtre glissante, l'apprentissage est alors continu mais coûteux,
- la seconde approche est similaire mais avec un décalage fixe ; lorsque la fenêtre glissante a été décalée d'un certain nombre d'instances, le modèle est ré-entraîné,
- la dernière approche est adaptative et repose sur les vecteurs de supports, définis entre autre comme les instances dont la norme est supérieure ou égale au rayon ; une instance qui n'est pas vecteur de support n'agit pas sur le modèle, donc le ré-apprentissage est réalisé lorsqu'une nouvelle instance est sur la surface de l'hyper-sphère ou à l'extérieur.

En remarquant l'importance du paramètre ν sur la précision des résultats, [Yao 2015] reprend l'approche adaptative de [Zhang 2009] en ajoutant la recherche optimale du paramètre ν. Cette recherche reprend l'algorithme OAQO proposé par [O'Reilly 2012] en simplifiant l'approche. La méthode résultante est moins précise

mais plus rapide à calculer (en évitant de générer un grand nombre de modèles).

En théorie, on a dans chaque noeud n instances d-variées. Dans [Shahid 2012a], la matrice $n \times d$ des données est découpée en $\lceil \frac{n}{d} \rceil$ matrices $d \times d$ dont on prend les transposées pour inverser les attributs et les instances. L'objectif est de représenter la distribution des attributs dans l'espace de représentation plutôt que celle des instances. Un modèle est alors appris pour chaque matrice $d \times d$ et les $\lceil \frac{n}{d} \rceil$ rayons résultants sont agrégés par la médiane. Ensuite, l'approche reprend celle de [Zhang 2008] avec un modèle de QSSVM normal et un second pour les attributs. [Shahid 2012b] pousse l'applicabilité de cette approche en temps réel avec plusieurs propositions pour un apprentissage adaptatif suivant les travaux de [Zhang 2009].

Représentation en ellipsoïdes centrées L'utilisation de la distance de Mahalanobis permet de tenir compte des dépendances entre variables. La logique d'utiliser cette distance pour la classification à une classe est la même que celle pour le clustering hyper-ellipsoïdal dans [Moshtaghi 2009].

Dans [Rajasegarar 2010], la méthode d'OCSVM à partir d'hyper-ellipsoïdes centrées (CESVM) est proposée et comparée aux QSSVM. Une approximation du calcul des $(\phi(\mathbf{x}_i) - \mu)\Sigma^{-1}(\phi(\mathbf{x}_i) - \mu)^T$ est également proposée pour (1) estimer la matrice Σ^{-1} pour éviter le coût d'inversion de Σ et (2) appliquer l'astuce du noyau puisqu'on ne souhaite pas calculer les $\phi(\mathbf{x}_i)$. L'approche pour la détection distribuée des anomalies reprend celle de [Rajasegarar 2007].

Considérant l'approche de [Rajasegarar 2010] pour modéliser les hyper-ellipsoïdes dans l'espace de représentation comme ayant une complexité trop élevée, [Zhang 2013b] propose de modéliser les hyper-ellipsoïdes dans l'espace d'origine. L'évaluation d'un nouveau point nécessite désormais de connaître la médiane (utilisée à la place de la moyenne) et la matrice de covariance de la distribution, en plus du rayon. Les modèles globaux agrègent également les médianes et les matrices de covariance. A partir de cette représentation, deux approches sont proposées :

— la première reprend la seconde approche de [Zhang 2009] avec un décalage dont l'amplitude est exactement la taille de la fenêtre,

— la seconde est une approche adaptative où la médiane et la matrice de covariance sont mises à jour en temps réel et le rayon est recalculé quand le taux d'anomalies dépasse un seuil ; lorsque le rayon d'un noeud est mis à jour, il y a également mise à jour du modèle parent.

2.2.7.3 Avantages et inconvénients

L'avantage des méthodes d'OCSVM est la capacité à générer des enveloppes non linéaires pour modéliser la normalité. Cependant, il en découle que la définition de l'anomalie est complexe et difficilement interprétable.

De plus, la résolution du problème d'optimisation est difficile à réaliser dans un noeud de faible puissance. C'est la raison pour laquelle les méthodes présentées ici se limitent aux cas de résolution linéaire. Il existe des méthodes reposant sur une résolution quadratique, que ce soit dans les flux de données [Krawczyk 2015,

Bhat 2020] ou dans les WSNs [Feng 2017]. Cependant, puisqu'elles ne proposent pas de solutions pour une mise à jour du modèle sans ré-apprentissage, le problème d'optimisation doit être résolu régulièrement ce qui rend ces approches peu viables.

2.2.8 Méthodes par reconstruction

La dernière catégorie de méthodes traitée est celle des méthodes par reconstruction, qui contiennent en particulier les méthodes spectrales qui sont présentées dans cette section.

2.2.8.1 Méthodes pour les réseaux de capteurs

Il existe deux types d'approches pour la détection d'anomalies par méthode spectrale selon la construction du modèle décrite dans notre taxonomie : l'approche par construction totale, utilisée dans [Chatzigiannakis 2006, Ahmadi Livani 2013], et l'approche par construction distribuée, mise en pratique par [Ahmadi Livani 2013]. Ces approches utilisent une analyse en composantes principales (ACP) pour modéliser les instances normales et détecter les anomalies.

Approche centralisée Dans l'approche de [Chatzigiannakis 2006], une topologie en clusters est supposée, avec un noeud de plus grande capacité à la tête de chaque cluster. Toutes les mesures sont transmises à la tête de cluster qui réalise un apprentissage centralisé et hors ligne. Cet apprentissage consiste à réaliser l'ACP à partir de la matrice de covariance des données de tout le cluster. En sortie, les composantes principales sont obtenues, et la règle de Kaiser [Jolliffe 2002] est appliquée pour en sélectionner un nombre réduit qui permettra la modélisation de la normalité.

La distinction entre les instances normales et les instances anormales se fait à travers la méthode des sous-espaces. L'idée est qu'une instance normale $\mathbf{x}$ peut être décomposée en deux portions $\mathbf{x} = \mathbf{x_{norm}} + \mathbf{x_{res}}$ où $\mathbf{x_{norm}}$ correspond à sa portion normale et $\mathbf{x_{res}}$ à sa portion résiduelle. L'ensemble des composantes principales retenues sont utilisées pour obtenir la composante normale d'un vecteur, et la décision sur son anormalité est réalisée à partir de l'erreur de prédiction quadratique, calculée à partir du carré de la norme de la composante résiduelle.

L'apprentissage étant particulièrement lourd, il n'est réalisé que lorsqu'il y a un changement important dans les coefficients de corrélations entre les différents attributs, qui quantifient leurs dépendances. Pour réaliser ce ré-apprentissage adaptatif, une fenêtre glissante est constituée des dernières mesures et étudie la variation des coefficients de corrélations.

Finalement, les informations sur les anomalies détectées et leur provenance est envoyée par les têtes de cluster à la station de base qui, ayant la connaissance de l'organisation complète du réseau, peut qualifier les anomalies comme des erreurs isolées ou des évènements d'intérêt.

[Ahmadi Livani 2013] propose une approche différente pour la détection d'anomalies. L'ACP est construite de la même manière, mais la mesure d'anomalie utilisée est la distance d'une instance à sa projection sur la composante principale. De plus, le modèle est mis à jour régulièrement plutôt que de manière adaptative.

Approche distribuée Dans [Ahmadi Livani 2013], une approche de construction distribuée du modèle est également proposée afin de réduire l'effort en communication dans le WSN. L'ACP nécessitant une normalisation des données, celle-ci doit maintenant être faite de manière distribuée en communiquant avec la tête de cluster qui calcule les paramètres de normalisation et les redistribuent aux autres noeuds.

Chaque noeud réalise un clustering de ses données et transmet à la tête de cluster les caractéristiques suffisantes de ses clusters. La détection d'anomalies calcule cette fois la distance non pas des instances mais des clusters, et en tenant compte de leur rayon, à leur projection sur la composante principale.

La partie la plus lourde de cette approche est le calcul distribué de l'ACP. Chaque noeud calcule d'abord la matrice centrée de ses données et transmet à la tête de cluster une matrice triangulaire obtenue par décomposition QR de cette matrice centrée. L'ACP peut être retrouvée de manière exacte à travers une décomposition QR de la combinaison des matrices triangulaires pour chaque combinaison de noeuds.

2.2.8.2 Avantages et inconvénients

Les méthodes spectrales détaillées ici tendent à fournir de bons résultats pour la détection d'anomalies. La construction du modèle est trop lourde pour un apprentissage continu qui n'est donc pas traité dans les méthodes citées. Cependant, il existe dans l'état de l'art diverses approches pour une construction itérative du modèle qui pourraient être appliquées [Li 2000, Zhao 2006, Deng 2016].

De plus, l'approche de construction totale force la communication de toutes les observations vers la tête de cluster. L'approche de construction limitée résout ce problème mais au prix d'une plus grande complexité calculatoire.

2.3 Bilan de l'état de l'art

La présentation des travaux appliquant les différentes catégories de méthodes de détection d'anomalies au problème des WSNs a permis de démontrer leur diversité, renforçant l'idée selon laquelle il peut être difficile de sélectionner une approche. L'objectif de cette section est de constituer un bilan de l'état de l'art en s'appuyant sur les Tableaux 2.2 et 2.3 qui décrivent les méthodes rencontrées selon la taxonomie proposée en introduction de ce chapitre, triées chronologiquement selon l'année de publication.

2.3.1 Quelques éléments sur la conception des tableaux

Les éléments de la taxonomie introduite, représentés par la Figure 2.1, sont déclinés dans les Tableaux 2.2 et 2.3 pour chaque méthode. Les valeurs possibles pour chaque élément ont été fournies dans la Section 2.1.2. Les abréviations décrites dans le Tableau 2.1 sont utilisées pour référencer ces valeurs, parfois accompagnées d'un élément entre parenthèses pour apporter une précision, comme c'est souvent le cas pour

L'approche pour la construction du modèle est souvent accompagnée d'une précision. Dans le cas d'une construction totale, il s'agit généralement de la portée des instances utilisées nécessaires à la construction du modèle. Si aucune précision n'est fournie, c'est que les instances de tous les noeuds du WSN sont nécessaires. Dans le cas d'une construction distribuée, la précision porte sur le type d'informations utilisé.

Les différentes dépendances sont représentées par des symboles. Le symbole "-" signifie que la méthode n'étudie pas ce type de dépendances. Le symbole "(X)" signifie une étude partielle mais pas dédiée, tandis que le symbole "X" témoigne d'une étude dédiée.

Un " ?" est utilisé dans le cas d'une incertitude sur la valeur lorsque l'article ne présente pas cet aspect. C'est notamment le cas pour la fréquence de mise à jour de certaines méthodes de classification basées sur les réseaux bayésiens où l'apprentissage n'est pas décrit ; il est alors possible qu'un ré-apprentissage total soit nécessaire pour les tables de probabilité plutôt qu'une simple mise à jour.

Enfin, lorsque la fréquence de mise à jour est désignée comme unique sans précision, il s'agit d'un ré-apprentissage régulier.

2.3.2 Un objectif clair : réduire la charge de communication du réseau

La question du coût en communications sonne comme un leitmotiv en parcourant l'état de l'art, souvent motivée par l'étude de [Zhao 2003] comparant le coût de communication à celui de calcul.

Dans les tableaux, cela se caractérise par :

— une construction des modèles majoritairement distribuée ou totale sur des voisinages restreints ; les instances n'ont alors pas à être communiquées dans tout le WSN,

— des mises à jour rarement continues des modèles qui ne sont pas locaux, ce qui forcerait théoriquement à communiquer vers un noeud central les mises à jour à la même fréquence que celle d'observation des instances, et serait donc équivalent en charge de communication au fait de communiquer toutes les instances,

— des approximations, qui peuvent avoir deux sources : (1) réduire la complexité des calculs ou (2) limiter les informations à transmettre dans le réseau.

Idéalement, pour pouvoir être utilisé dans le cadre des WSNs, un modèle de détection d'anomalies doit donc pouvoir être construit de manière distribuée avec des modèles locaux ou des caractéristiques suffisantes des données. Souvent, cette caractéristique s'accompagne d'une capacité à mettre à jour le modèle sans nécessiter de ré-apprentissage complet, en ajoutant simplement de nouveaux fragments du modèle.

2.3.3 Priorité sur les dépendances spatiales

A la différence des méthodes appliquées aux flux de données, qui n'ont pas à traiter l'aspect spatial, les méthodes appliquées aux WSNs se concentrent principalement sur le traitement des dépendances spatiales.

La distribution des données dans l'environnement est presque toujours considérée comme homogène ou spatiale (selon la taxonomie décrite en Section 2.1.2). La dépendance spatiale est alors traitée en tenant compte des autres noeuds du réseau dans le premier cas ou des noeuds proches dans le second. Même dans le cas où la distribution serait comportementale, une topologie du réseau en clusters pour cibler le comportement permettrait de le traiter comme dépendance spatiale en étudiant les noeuds du même cluster.

En pratique, là où les dépendances temporelles ou entre attributs ne sont souvent traitées que partiellement (en considérant respectivement un entraînement sur un ensemble d'instances avec des temporalités différentes ou un modèle multivarié), les dépendances spatiales sont systématiquement étudiées. En effet, on retrouve une majorité de modèles globaux, représentés généralement par un apprentissage centralisé.

Puisque la génération d'un modèle global dans le WSN semble suffisante pour traiter des dépendances spatiales, l'intérêt de l'état de l'art pour ce type de dépendances peut se traduire par la nécessité d'une capacité des modèles à être combinés. De plus, la différenciation entre les erreurs et les évènements d'intérêt est souvent réalisée à travers la comparaison entre les modèles locaux et globaux. Si la propriété de combinaison est vérifiée, il est d'autant plus logique de conserver ces deux types de modèles pour comparaison.

Cependant, traiter directement les dépendances spatiales nécessite des connaissances sur la topologie du réseau ou la distribution des données dans l'espace. De ce fait, elles ne sont pas traitées dans cette thèse, mais nous travaillons avec des méthodes composables qui permettent le traitement de ces dépendances une fois les connaissances requises introduites.

2.3.4 Dissensions avec l'état de l'art de la détection d'anomalies dans les flux de données

Il est assez surprenant de voir la proportion de méthodes dédiées aux WSNs nécessitant un ré-apprentissage complet des modèles, en particulier dans les articles les plus récents. C'est d'autant plus étonnant que le besoin d'apprentissage continu

est décrit comme une spécifité des flux de données.

A travers l'état de l'art parcouru, les méthodes appliquées aux flux de données sont néanmoins souvent en décalage par rapport à celles appliquées aux WSNs. Si l'objectif de limitation des communications et l'importance des dépendances spatiales peuvent être des explications à ce phénomène, il serait cependant pertinent pour les WSNs d'utiliser davantage l'état de l'art de la détection d'anomalies dans les flux de données.

Parmi les spécificités des flux de données, on retrouve notamment la question de la rapidité des méthodes pour traiter les instances dès qu'elles sont générées. Cette question parait pertinente dans le cadre des WSNs, en particulier pour les méthodes locales avec mises à jour continues, mais elle n'est pourtant pas soulevée.

2.3.5 Une définition de l'anomalie fixée

Les méthodes de l'état de l'art fixent généralement une "définition" de l'anomalie particulière. Cette définition mathématique, qui n'est pas toujours explicite, est intimement liée à la catégorie des méthodes employées. Dans le cas de méthodes basées sur la distance par exemple, [Palpanas 2003] et [Subramaniam 2006] définissent explicitement une anomalie comme une instance ayant moins d'un certain nombre de voisins dans un rayon R.

Les anomalies ainsi définies sont ensuite recherchées de manière exacte ou approchée. Dans le cas approché, la différence avec le résultat exact attendu constitue une bonne mesure d'évaluation de la précision des méthodes.

2.3.6 Justification des verrous posés

L'état de l'art réalisé dans ce chapitre permet finalement de préciser les verrous identifiés dans la Section 1.5.3, où l'objectif de cette thèse a été formulé comme :

Objectif 1 *Proposer une solution,* automatisable *et* agnostique au contexte industriel*, à la problématique de détection d'anomalies dans les réseaux de capteurs.*

Selon les conclusions tirées dans cette section, les solutions réalisant cet objectif doivent soutenir les propriétés suivantes :

- les modèles de détection d'anomalies doivent pouvoir *être combinés*, à la fois pour limiter la charge de communication dans le WSN où ils seront à terme déployés et pour permettre, à l'abenir, de tenir compte de possibles dépendances spatiales,
- les dépendances d'attributs et temporelles doivent à minima être observées partiellement à travers *des modèles (1) multivariés et (2) construits sur des plages temporelles larges*,
- pour chaque méthode, *une définition de l'anomalie doit être fixée* ; dans un cadre non supervisé, cette définition peut constituer une forme de labellisation à laquelle se raccrocher pour l'évaluation,

— il semble important, dans les solutions proposées, de *se rapprocher de l'état de l'art de la détection d'anomalies dans les flux de données* pour tenir compte de leurs spécificités.

L'état de l'art a par ailleurs montré que les méthodes applicables à la détection d'anomalies dans les WSNs sont nombreuses et variées. De plus, dans les articles parcourus, très peu d'hypothèses fortes sont faites sur les contextes d'application et qui permettraient de sélectionner une méthode adaptée à un contexte particulier. Cela justifie le Verrou 1 défini dans le chapitre précédent concernant la sélection des méthodes :

Verrou 1 *La sélection de méthodes de détection d'anomalies adaptées aux données traitées parmi le large spectre de méthodes non supervisées de l'état de l'art.*

Ce verrou motive la conception d'un "environnement de travail" permettant d'appréhender la problématique de cette thèse, réalisée dans le Chapitre 3 sous la forme d'un cadre opérationnel pour la détection d'anomalies reposant sur des définitions spécifiques de l'anomalie.

Le second verrou défini dans le Chapitre 1 concerne le paramétrage des méthodes. L'état de l'art n'a pas mis en avant de méthodes sans paramètres, ce verrou est donc redéfini ainsi :

Verrou 2 *La suppression, ou au moins la simplification, de la phase de paramétrage de la solution, et donc des méthodes qui la composent.*

Le Verrou 3 concerne la satisfaction des contraintes des WSNs et peut être précisé, d'après l'état de l'art présenté dans ce chapitre, de la manière suivante :

Verrou 3 *Une méthode applicable aux réseaux de capteurs doit être composable, satisfaire les contraintes des flux de données et, a minima, tenir compte des dépendances d'attributs à travers une étude multivariée et des dépendances temporelles à travers un apprentissage en ligne sur des plages temporelles suffisamment larges.*

Ce troisième verrou est traité à la fois dans le premier cas d'application du cadre opérationnel, proposé dans le Chapitre 3, et dans les méthodes proposées dans les Chapitres 5 et 6.

Abréviation	Valeur
Méthodes	
S.p.	Statistiques paramétriques
S.n.p.	Statistiques non-paramétriques
D.	Distance
D.l.	Densité locale
Clus.	Clustering
Class.r.b.	Classification avec réseaux bayésiens
Class.OC	Classification à une classe
R.	Reconstruction
Localisation (apprentissage ou détection)	
C.	Centralisé
L.	Localisé
Construction	
Distr.	Distribuée
Fréquence de mise à jour	
Cont.	Continue
Rég.	Régulière
Adap.	Adaptatative
Topologie	
Hiér.	Hiérarchique
Clust.	en Clusters
Vois.	Voisinage
Homog.	Homogène
Distribution	
Spat.	Spatiale
Comport.	Comportementale
Dépendances	
T	Temporelles
S	Spatiales
A	d'Attributs

TABLE 2.1 – Abréviations utilisées dans les Tableaux 2.2 et 2.3 pour décrire la taxonomie de la Section 2.1.2.

TABLE 2.2 – Récapitulatif des méthodes de l'état de l'art selon la taxonomie proposée (Partie 1)

Méthode	Catégorie	Localisation		Construction	Fréquence	Topologie	Distribution	Dépendances			Approx.
		Apprent.	Détection					T	S	A	
[Palpanas 2003]	S.n.p. + D.	L. + C.	L. + C.	Distr. (KDE)	Cont.	Hiér.	Homog.	(X)	X	(X)	Oui
[Elnahrawy 2004] (voisinage)	Class.r.b.	L.	L.	Totale (voisins)	Cont.	Vois.	Spat.	X	X	-	Oui
[Elnahrawy 2004] (hiérarchie)	Class.r.b.	C.	L.	Distr. (histog.)	Rég.	Vois. + Hiér.	Homog.	X	X	-	Oui
[Subramaniam 2006] (dist.)	S.n.p. + D.	L. + C.	L. + C.	Distr. (KDE)	Rég.	Hiér.	Homog.	(X)	X	(X)	Oui
[Subramaniam 2006] (d. loc.)	S.n.p. + D.l.	L. + C.	L.	Distr. (KDE)	Rég.	Hiér.	Homog.	(X)	X	(X)	Oui
[Branch 2006] [Branch 2013]	D.	L. (global)	L.	Distr. (ens. supports)	Cont.	Vois.	Homog.	(X)	X	(X)	Non
[Rajasegarar 2006] [Rajasegarar 2014]	Clust.	L. + C.	L.	Distr. (clust.)	Unique	Hiér.	Homog.	(X)	X	(X)	Oui
[Janakiram 2006]	Class.r.b.	L.	L.	Totale (voisins)	Adapt. (?)	Vois.	Spat.	X	X	X	Oui
[Chatzigiannakis 2006]	R.	C.	C.	Totale (clust.)	Unique (adapt.)	Clust.	Spat.	(X)	X	X	Non
[Bettencourt 2007] (p.)	S.p.	L.	L.	Totale (voisins)	Cont.	Vois.	Spat.	X	X	-	Oui
[Bettencourt 2007] (n.p.)	S.n.p.	L.	L.	Totale (voisins)	Cont.	Vois.	Spat.	X	X	-	Oui
[Wu 2007]	S.p.	L.	L.	Totale (voisins)	Unique (cont.)	Vois.	Homog.	-	X	-	Oui
[Sheng 2007]	D.	C.	L.	Distr. (histog.)	Unique	Hiér.	Homog.	(X)	X	-	Non
[Zhang 2007]	D.	L. + C.	L.	Distr. (ens. supports)	Cont.	Hiér.	Homog.	(X)	X	(X)	Non

TABLE 2.3 – Récapitulatif des méthodes de l'état de l'art selon la taxonomie proposée (Partie 2)

Méthode	Catégorie	Localisation		Construction	Fréquence	Topologie	Distribution	Dépendances			Approx.
		Apprent.	Détection					T	S	A	
[Rajasegarar 2007]	Class.OC	L. + C.	L.	Distr. (rayons)	Unique	Hiér. ou clust.	Homog.	(X)	X	(X)	Non
[Moshtaghi 2009]	Clust.	L. + C.	L.	Distr. (clust.)	Unique	Hiér.	Comport.	(X)	X	X	Non
[Zhang 2009]	Class.OC	L.	L.	Distr. (rayons)	Unique (cont. ou rég. ou adapt.)	Vois.	Spat.	(X)	X	(X)	Non
[Rajasegarar 2010]	Class.OC	L. + C.	L.	Distr. (rayons)	Unique (adapt.)	Hiér. ou clust.	Homog.	(X)	X	X	Oui
[Shahid 2012a] [Shahid 2012b]	Class.OC	L.	L.	Distr. (rayons)	Unique (adapt.)	Vois.	Spat.	(X)	X	X	Non
[Xie 2013]	D.	C.	L.	Distr. (hyper-cubes)	Unique	Clust.	Spat.	(X)	X	(X)	Oui
[Zhang 2013b]	Class.OC	L.	L.	Distr. (rayons)	Unique (adapt.)	Vois.	Spat.	(X)	X	X	Non
[Ahmadi Livani 2013] (totale)	R.	C.	C.	Totale (clust.)	Unique	Clust.	Spat.	(X)	X	X	Non
[Ahmadi Livani 2013] (distrib.)	R.	C.	C.	Distrib. (clust.)	Unique	Clust.	Spat.	(X)	X	X	Non
[Titouna 2015]	Class.r.b.	L. + C.	C.	Totale (clust.)	Cont. (?)	Vois. + clust.	Spat.	X	X	(X)	Oui
[Yao 2015]	Class.OC	L.	L.	Distr. (rayons)	Unique (adapt.)	Vois.	Spat.	(X)	X	(X)	Non
[Abid 2017]	Clust.	C.	C.	Totale	Unique	Hiér.	Homog.	(X)	X	(X)	Non
[Titouna 2019]	Class.r.b.	C.	L.	Totale	Unique	Hiér.	Homog.	X	(X)	-	Oui

Chapitre 3

Cadre opérationnel unifié pour la détection d'anomalies et son évaluation non supervisée

Le chapitre précédent a permis de montrer la diversité des méthodes applicables aux WSNs et l'a identifiée comme une barrière à l'approche du domaine. Une des problématiques de cette thèse est en effet le choix *automatique* de méthodes génériques. Pour traiter cette problématique, il est nécessaire de réduire le nombre de méthodes utilisables à travers des critères de sélection spécifiques.
Ainsi, ce troisième chapitre propose un cadre opérationnel pour traiter le problème de la détection d'anomalies dans les réseaux de capteurs. Ce cadre opérationnel, baptisé *WOLF* pour *WSNs OutLier detection Framework*, prend la forme d'une boîte à outils constituée d'un ensemble de définitions différentes de l'anomalie, chacune accompagnée de méthodes appropriées pour détecter les anomalies associées. De plus, une approche pour l'évaluation non supervisée associée à WOLF est proposée. Enfin, les critères liés aux définitions et méthodes sont introduits pour permettre d'enrichir WOLF, et un cadre opérationnel unifié, *WOLF-KDE*, est développé.

Sommaire

3.1 Présentation du cadre opérationnel

Cette première section reprend tout d'abord les motivations derrière la conception d'un tel cadre opérationnel, en s'appuyant sur les observations mises en avant dans les chapitres précédents, avant de décrire les choix de conception retenus. Une troisième sous-section est dédiée à l'approche d'évaluation de la précision des méthodes au sein de ce cadre.

3.1.1 Motivations

3.1.1.1 Sélection automatique de méthodes génériques

Dans la Section 1.5.1, la problématique industrielle a été formalisée, conduisant à la nécessité de développer une solution automatisable et agnostique ; en effet, les anomalies doivent pouvoir être détectées dans n'importe quel cadre sans nécessiter l'intervention d'un expert métier.

Or, pour détecter des anomalies, une solution doit s'appuyer sur l'utilisation d'une ou plusieurs méthodes de détection. Pour qu'elle soit automatisable et agnostique, la sélection des méthodes, mais aussi de leurs paramètres, doit également l'être, et ce pour n'importe quel contexte d'étude.

L'analyse de l'état de l'art réalisée dans le Chapitre 2 a cependant démontré l'existence d'un nombre important d'approches pouvant être appliquées au problème de détection d'anomalies dans les WSNs. Réduire le nombre de méthodes utilisables à travers la conception d'un cadre bien délimité pourrait constituer une solution à ce problème.

3.1.1.2 Problème de l'évaluation non supervisée

La Section 1.4 a également présenté les limites de l'évaluation des méthodes dans un cadre non supervisé. Il n'est alors pas possible de comparer simplement les performances de différentes approches pour ne sélectionner que la meilleure, ce qui renforce la problématique de sélection automatique d'une méthode.

La solution d'évaluation externe (reposant donc sur des labels) usuelle consiste à réaliser un benchmark de l'ensemble des méthodes sur un ensemble de jeux de données représentatifs du problème de détection d'anomalies abordé. Dans l'état de l'art, le jeu de données public le plus utilisé pour évaluer les performances des méthodes dans le cadre des WSNs est celui de l'Intel Berkeley Research Lab (IBRL) [1]. Cependant, un seul jeu de données est insuffisant, et une approche similaire à celle de [Lavin 2015] avec des données de WSNs serait nécessaire.

La conception d'un cadre opérationnel avec des hypothèses suffisamment fortes sur les méthodes peut néanmoins contourner ce problème avec une forme d'évaluation interne, ne reposant que sur les données et les hypothèses retenues.

1. Voir `http://db.csail.mit.edu/labdata/labdata.html`.

3.1.1.3 Une catégorisation et des définitions

La Section 2.3.5 a mis l'accent sur des "définitions" de l'anomalie parfois précisées par les méthodes de l'état de l'art. Ces définitions peuvent être perçues comme des hypothèses fortes sur lesquelles s'appuyer pour l'évaluation de la précision des méthodes. En effet, elles définissent le résultat attendu, et donnent donc accès à une forme de labellisation. A titre d'exemple, les (k, R)-anomalies de [Knorr 1998] constituent une de ces définitions, et fournissent le résultat exact attendu ; toutes les instances ayant moins de k éléments dans leur voisinage à R sont des anomalies.

L'avantage d'utiliser de telles définitions est également de faciliter l'interprétation du résultat en aval par un expert puisque la logique sous-jacente aux résultats renvoyés est maîtrisée.

Un environnement s'appuyant sur ces définitions permettrait à la fois de répartir les méthodes et de les comparer sans utiliser de jeux de données labellisés.

3.1.2 Description de l'environnement

3.1.2.1 Structure de boîte à outils

Le cadre opérationnel proposé, nommé *WOLF* pour *WSNs OutLier detection Framework*, repose sur une structure de boîte à outils ; en effet, WOLF est organisé en différents "compartiments" qui correspondent à un besoin, à savoir une définition spécifique de l'anomalie, et dans chacun de ces compartiments se trouvent les "outils" répondant au besoin, à savoir des méthodes pour traiter cette définition de l'anomalie. La construction d'un tel cadre opérationnel nécessite de fixer plusieurs définitions de l'anomalie et d'implémenter des méthodes associées à ces différentes définitions.

Notons cependant que l'utilisation de la boîte à outils implique en théorie de pouvoir automatiquement déterminer le besoin pour sélectionner les méthodes dans le bon compartiment. Le choix retenu dans cette thèse, et qui sera poussé dans le chapitre suivant, est de ne pas fixer une unique définition mais de considérer qu'elles doivent toutes être appliquées, permettant de dire "cette instance est anormale selon telle définition mais normale selon telle autre".

Enfin, comme mentionné dans la sous-section précédente, il est également nécessaire de restreindre le nombre de méthodes. Pour ce faire, un ensemble de critères doivent être formalisés à la fois sur la définition d'anomalie et sur les méthodes.

3.1.2.2 Formalisation d'un critère sur la définition d'anomalie

Rappelons d'abord que la définition générale de l'anomalie, proposée en Section 1.2.1.1 (Déf. 1), introduisait une notion de "concept de normalité". C'est ce concept de normalité qui est associé à la notion de "définition" utilisée dans ce troisième chapitre. Aussi, pour être suffisamment interprétable, ce concept de normalité doit être relativement simple.

Un seul critère est ainsi formalisé sur la définition d'une anomalie pour qu'elle puisse être intégrée à WOLF : celle-ci doit pouvoir être écrite sous une forme simple n'impliquant qu'une notion de *seuil* sur une *métrique* associée à chaque instance et donnée par la définition. Formellement, cela signifie qu'elle doit prendre la forme : "une instance est une anomalie si sa *métrique associée* est inférieure (ou supérieure) à un *seuil*".

Pour reprendre l'exemple des (k, R)-anomalies de [Knorr 1998], la métrique est le nombre d'instances dans son voisinage à R et le seuil est k. On obtient la définition : "une instance est une anomalie si son nombre de voisins à R est inférieur à k".

3.1.2.3 Formalisation de critères sur les méthodes

Les critères sur les méthodes applicables proviennent des spécificités des réseaux de capteurs ou des flux de données.

Critères provenant des réseaux de capteurs Comme observé dans le chapitre précédent, la principale spécificité introduite par les WSNs est le besoin de réduire la charge de communication. Celle-ci est associée à la possibilité de distribuer le calcul en combinant des modèles. Les méthodes doivent ainsi respecter deux critères :

— les modèles générés doivent pouvoir être combinés,
— le nombre de paramètres constituant les modèles doit être inférieur au nombre d'instances le composant pour que la charge de communication induite par la transmission des modèles soit inférieure à celle induite par la transmission de toutes les instances.

Critères provenant des flux de données Les différentes spécificités des flux de données ont été fournies en Section 1.3.2. Parmi ces spécificités, retenons principalement que les méthodes doivent traiter les instances rapidement et que l'entièreté du jeu de données ne peut être conservé en mémoire. Aussi, deux critères sont à respecter :

— les méthodes doivent avoir un temps d'exécution faible,
— elles ne doivent pas nécessiter de stocker toutes les instances en mémoire, et plus précisément la taille du modèle ne doit pas être dépendante du nombre total d'instances.

3.1.2.4 Choix des définitions et des méthodes

Dans l'état de l'art présenté, des définitions spécifiques ont été fournies dans le cas des anomalies basées sur la distance et sur la densité locale [Palpanas 2003, Subramaniam 2006, Sheng 2007]. Ces définitions étant également facilement interprétables, elles font de bonnes candidates pour WOLF. A cela peuvent s'ajouter les anomalies statistiques dont il est également facile de fournir une définition selon un critère formalisé.

Anomalies statistiques Les anomalies statistiques sont "des instances ayant une faible probabilité d'occurence", ce qui peut être redéfini pour correspondre à notre formalisme, et de manière générique, comme : "*une instance est une anomalie si sa probabilité d'occurrence est faible*". Pour préciser la définition, il est nécessaire de fixer la métrique, à savoir la probabilité d'occurrence d'une instance, et le seuil, à savoir à partir de quelle valeur la probabilité est considérée comme faible. Plusieurs méthodes correspondant à cette définition ont été rencontrées dans l'état de l'art fourni :

- parmi les méthodes statistiques paramétriques, seules les GMM utilisées dans [Yamanishi 2004], appliquées aux flux de données, sont utilisables en multivarié tout en respectant les critères formalisés sur les méthodes,
- parmi les méthodes statistiques non-paramétriques, les méthodes estimant la densité à partir de KDE [Kristan 2011, Yamanishi 2004] sont utilisables selon les critères formalisés, mais l'utilisation d'histogrammes [Goldstein 2012] impliquerait un trop grand nombre de paramètres en mémoire, surtout dans un cadre multivarié,
- les méthodes de classification à base de réseaux bayésiens [Elnahrawy 2004, Janakiram 2006] suivent également la définition statistique proposée, cependant leur construction et leur maintien est plus lourd encore que les méthodes à base d'histogrammes.

Anomalies de distance Dans le cas des anomalies de distance, on peut proposer deux définitions génériques : "*une instance est une anomalie si ses voisins sont suffisamment éloignés d'elle*", où la métrique repose sur une distance, et "*une instance est une anomalie si elle possède peu d'éléments dans son voisinage*", où la métrique repose sur un dénombrement. Dans les deux cas, il faut préciser le voisinage et un seuil. Les méthodes correspondant à cette définition appartiennent à la catégorie des méthodes basées distance et sont soit exactes [Sheng 2007, Branch 2006, Zhang 2007, Kontaki 2011] soit approchées [Palpanas 2003, Subramaniam 2006, Xie 2011].

Anomalies de densité locale Dans ce cas de figure, la définition nécessite de fixer une métrique locale et se traduit comme : "*une instance est une anomalie si la valeur de sa métrique locale est supérieure (ou inférieure) à un seuil*". Dans le chapitre précédent, deux de ces métriques ont été introduites :

- le LOF, utilisé dans les flux de données, avec iLOF [Pokrajac 2007], I-IncLOF [Karimian 2012], MiLOF [Salehi 2016], DILOF [Na 2018] et TA-DILOF [Huang 2020],
- le MDEF, utilisé dans les WSNs [Subramaniam 2006].

Application des trois définitions Comme mentionné en décrivant la structure en boîte à outils, et afin de ne pas avoir à fixer automatiquement une définition, le choix retenu pour cette thèse est de toutes les utiliser. En théorie, ce choix revien-

drait à avoir un modèle en mémoire par définition, ce qui pourrait être lourd au sein des noeuds capteurs s'ils ne peuvent stocker qu'un faible nombre de paramètres.

Cependant, les modèles de KDE sont utilisés par des méthodes pour les trois définitions choisies [Kristan 2011, Yamanishi 2004, Palpanas 2003, Subramaniam 2006]. Il est donc possible de mutualiser l'occupation mémoire pour ces méthodes avec un unique modèle permettant de détecter les anomalies correspondant à toutes ces définitions. Un cadre unifié, nommé *WOLF-KDE*, est donc proposé. Il repose sur les trois définitions mentionnées et associées à trois méthodes basées sur des modèles de KDE. Dans la suite de ce chapitre, WOLF-KDE est décrit et comparé à d'autres méthodes de référence pour les trois définitions.

3.1.3 Évaluation non supervisée de la précision

L'utilisation de définitions formelles permet d'évaluer les méthodes au sein de WOLF ; il est effectivement possible de générer des labels en se reposant uniquement sur la définition retenue et d'évaluer ensuite les méthodes à partir de ces labels. On considère alors qu'une méthode est performante si elle est conforme à la définition qu'elle suit. Dans ce cas, les méthodes exactes peuvent être considérées comme "parfaites" en termes de précision.

Cependant, lorsqu'on évalue des méthodes approchées, et parce qu'on s'intéresse également au score d'anomalie renvoyé, le plus important n'est pas de déterminer exactement les anomalies au travers d'un seuil mais plutôt de classer correctement les instances selon leur degré d'anormalité. Pour évaluer la précision d'un modèle, l'approche d'évaluation proposée dans cette sous-section compare le classement des instances induit par le modèle au classement induit par la définition de l'anomalie suivie. Dans l'exemple des (k, R)-anomalies, un tel classement serait réalisé selon le nombre d'éléments dans le voisinage à R des instances.

Pour évaluer la distance entre deux classements, une des mesures les plus intuitives est celle de la règle de Spearman [Spearman 1987]. Supposons le classement de référence induit par les indices $i = \{1, ..., n\}$ de n éléments à classer et π une permutation de ces indices où $\pi(i)$ correspond à la position de l'élément i dans le second classement. La distance entre les classements est alors définie par :

$$F(\pi) = \sum_{i=1}^{n} |i - \pi(i)| \,. \tag{3.1}$$

Cette distance est généralisée par [Kumar 2010] pour tenir compte des différents poids :

- le poids des éléments : le bon classement de certains éléments peut être plus important que d'autres, il est donc nécessaire de pouvoir les pondérer,
- le poids des positions : parfois, bien classifier les premiers éléments est plus important que les derniers, ou vice-versa,
- le poids de la similarité entre les éléments : échanger les positions dans le classement de deux éléments très similaires est moins grave qu'échanger la position d'éléments très différents.

Nous proposons ci-dessous la méthode d'évaluation non supervisée *WOLF-Eval*, au sein du cadre opérationnel WOLF, qui adapte la règle de Spearman au cas de l'évaluation de classements induits par des méthodes de détection d'anomalies. Dans ce cas, on peut considérer qu'il est plus important de bien positionner les anomalies que les instances normales. Ainsi, la pondération des éléments peut être faite à partir du score d'anomalie induit par la définition, noté w_i pour la i-ième instance. Mais il est également important de considérer la similarité entre les éléments, ou plus précisément la distance entre les instances, notée $D_{i,j}$ pour les instances i et j. Selon [Kumar 2010], la forme généralisée de la distance par la règle de Spearman est alors :

$$F_g(\pi) = \sum_{i=1}^{n} w_i \cdot \left| \sum_{j|j\leq i} w_j D_{i,j} - \sum_{j|\pi(j)\leq\pi(i)} w_j D_{i,j} \right|. \tag{3.2}$$

Pour faciliter la comparaison des résultats entre les différentes applications, le score est normalisé et on prend comme poids $\bar{w_i} = w_i / \sum_{j=1}^{n} w_j$.

La définition de la méthode d'évaluation WOLF-Eval constitue finalement une des forces de WOLF. Pour sa mise en place, il est nécessaire de déterminer :

— le classement des anomalies selon la définition retenue,
— les scores d'anomalie selon la définition retenue, utilisés pour la pondération,
— le classement des anomalies selon la méthode évaluée.

3.2 Développement des définitions dans un cadre unifié

Dans cette section, les trois définitions retenues précédemment sont développées, puis les méthodes pour le cadre unifié WOLF-KDE sont décrites avec des méthodes de référence auxquelles les comparer. Enfin, la précision est évaluée par l'approche d'évaluation non supervisée proposée dans la Section 3.1.3 et les résultats sont restitués.

3.2.1 Définition statistique

3.2.1.1 Définition de l'anomalie

Une méta-définition pour les anomalies statistiques est :

Définition 4 *Une anomalie statistique est une instance ayant une probabilité d'occurrence inférieure à un seuil.*

Comme mentionné dans la Section 3.1.2.4, cette définition nécessite de fixer deux éléments : la probabilité d'occurrence d'une instance et l'interprétation d'une faible probabilité par un seuil. Notons que l'évaluation des méthodes à travers la distance entre les classements ne tient pas compte du seuil fixé.

Sur une variable aléatoire continue, qui correspond à un attribut continu, la notion de probabilité n'a cependant pas de sens pour une valeur isolée, ce qui correspond à une instance.

Soit X une telle variable en une dimension, on définit la probabilité sur un intervalle $[a, b]$ en utilisant la pdf f associée à X :

$$P\left(a \leq X \leq b\right) = \int_a^b f\left(u\right) du\,. \tag{3.3}$$

Aussi, on a nécessairement $P\left(X = a\right) = \int_a^a f\left(u\right) du = 0$.

Plus précisément, en p dimensions, on définit une variable aléatoire multivariée, ou vecteur aléatoire, $X = (X_1, X_2, ..., X_p)$. La probabilité est alors appliquée à un produit de p intervalles correspondant à chaque dimension, $[a_1, b_1] \times [a_2, b_2] \times ... \times [a_p, b_p]$, ce qui est un élément de la tribu Borélienne de $\mathbb{R}^p$. La probabilité s'écrit dans ce cas :

$$P\left(X \in [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_p, b_p]\right) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \ldots \int_{a_p}^{b_p} f\left(u_1, u_2, \ldots, u_p\right) du_1 du_2 \ldots du_p\,. \tag{3.4}$$

A partir de cette observation, deux approches peuvent être envisagées pour évaluer la "probabilité d'occurence" d'une instance :

— calculer la probabilité dans un intervalle autour de l'instance, ce qui est réalisé dans l'état de l'art par les méthodes de classification à base de réseaux bayésiens,

— utiliser la pdf qui est, quant à elle, définie sur $\mathbb{R}^p$ et non sur une tribu de $\mathbb{R}^p$.

Comme précisé dans la Section 3.1.2.4, le cadre appliqué WOLF-KDE unifie les modèles en reposant uniquement sur des KDE, et l'estimation de la pdf est donc retenue puisqu'elle est possible par KDE, donnant la définition plus précise :

Définition 5 *Une anomalie statistique est une instance pour laquelle l'évaluation de la pdf est inférieure à un seuil.*

3.2.1.2 Estimation de la pdf par KDE

La méthode d'estimation de la pdf par KDE a déjà été présentée dans le Chapitre 2, en Section 2.2.2.1. Cette estimation utilise des fonctions noyau centrées sur différents échantillons de la distribution avec l'intuition d'apporter une forme de continuité aux méthodes d'estimation par histogrammes.

Cette approche a l'avantage de ne pas supposer une distribution spécifique des données. De plus, les modèles peuvent facilement être combinés ; il suffit de combiner les échantillons qui les définissent, soit via leur union, soit via un sous-échantillonnage de leur union, et de combiner les variances utilisées pour le calcul de la largeur de bande par la règle de Scott [Scott 1992].

A des fins de comparaison, et pour conforter le choix de KDE, une seconde méthode permettant de mettre en oeuvre la Définition 5 est évaluée : l'estimation de la pdf par GMM.

3.2.1.3 Estimation par modèle de mélange gaussien

Alors que l'approche par KDE estime la pdf à partir d'un ensemble d'échantillons et de la largeur de bande, l'approche par GMM est un modèle qui mélange k modèles gaussiens paramétrés par un ensemble de couples (μ_i, Σ_i) et leurs poids associés c_i, $1 \leq i \leq k$. Précisément, la densité pour un mélange gaussien est donnée par :

$$f(\mathbf{x}) = \sum_{i=1}^{k} c_i p(x|\mu_i, \Sigma_i) \tag{3.5}$$

avec :

$$p(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \tag{3.6}$$

et $\sum_{i=1}^{k} c_i = 1$.

Pour estimer la pdf, il faut alors estimer les (c_i, μ_i, Σ_i) pour un k fixé. Cette estimation est réalisée à travers une procédure d'optimisation en utilisant le critère de maximum de vraisemblance. En pratique, l'approche la plus utilisée est la procédure itérative espérance-maximisation proposée par [Neal 1998], et adaptée dans SmartSifter [Yamanishi 2004] pour une application dans les flux de données où chaque itération de la phase d'optimisation correspond à l'observation d'une nouvelle instance.

Bien qu'il s'agisse d'une approche paramétrique, n'importe quelle distribution peut théoriquement être convenablement estimée par des GMM pour k correctement fixé. De plus, pour des performances équivalentes, il faut généralement un nombre d'échantillons pour un modèle par KDE bien supérieur au nombre k de gaussiennes grâce à la phase d'optimisation, résultant en une réduction du nombre de paramètres.

3.2.1.4 Évaluation de l'estimation

Description de l'expérimentation Pour évaluer les performances des modèles avec WOLF-Eval, il est nécessaire de calculer le classement théorique des instances et leur score d'anomalie théorique. Pour ce faire, les données de test utilisées sont celles de mélanges gaussiens dont la densité réelle est connue et donnée par l'Équation 3.5. Le score w_i d'une instance $\mathbf{x}_i$ est donné par $w_i = 1/(1 + f(\mathbf{x}_i))$. Ce score est ensuite normalisé et utilisé comme poids $\bar{w}_i = w_i / \sum_{j=1}^{n} w_j$ dans le calcul de la distance entre les classements, comme défini dans l'Équation 3.2.

Notons également que le score utilisé pour le classement théorique n'est pas généré dans une fenêtre glissante, comme celui des méthodes évaluées, mais en tenant compte de l'entièreté du jeu de données. Ceci ne doit pas avoir d'impact étant donné la stationnarité du jeu de données et permet de réduire le temps de calcul de WOLF-Eval.

A titre indicatif, une distance de référence est fournie sur chacun des graphiques restitués ; celle-ci est obtenue en prenant la moyenne des distances obtenues, selon WOLF-Eval, pour 1000 classements générés aléatoirement.

Afin d'évaluer l'influence du nombre de gaussiennes et du nombre de variables sur le résultat, l'évaluation est réalisée avec :

— une distribution avec une seule gaussienne (centrée réduite), en dimensions 2, 3 et 4 ;

— une distribution avec huit gaussiennes en dimensions 2 et 3.

De plus, parce que les jeux de données sont générés aléatoirement, chaque expérience est répétée 10 fois pour observer l'écart-type des résultats entre les expériences successives.

Paramétrage de la méthode par KDE Plusieurs paramètres doivent être fixés pour la méthode par KDE :

— la méthode pour fixer la largeur de bande : la règle de Scott est utilisée et définit la largeur de bande $\mathbf{H}$ comme

$$\mathbf{H}_{i,j} = \begin{cases} 0 & \forall i \neq j, \\ \sqrt{5} n^{-\frac{1}{p+4}} \sigma_i & \text{sinon}, \end{cases} \tag{3.7}$$

avec σ_i l'écart-type du i-ième des p attributs et n le nombre d'échantillons utilisés comme centres de noyaux ;

— la fonction noyau à utiliser : bien que le noyau gaussien serait plus intéressant pour une comparaison avec les GMM, le noyau retenu est celui d'Epanechnikov pour maintenir une cohérence avec les définitions d'anomalies de distance et de métriques locales. Dans le cas multivarié, et en fixant $\mathbf{H}$ via la règle de Scott, le noyau d'Epanechnikov s'écrit :

$$\mathbf{K_H}(\mathbf{x}) = \begin{cases} \left(\frac{3}{4}\right)^d \frac{1}{|\mathbf{H}|} \prod_{i=1}^{p} \left(1 - \left(\frac{\mathbf{x}_i}{\mathbf{H_{i,i}}}\right)^2\right) & \text{si} \quad \forall i, \frac{\mathbf{x}_i}{\mathbf{H_{i,i}}} < 1 \\ 0 & \text{sinon} ; \end{cases} \tag{3.8}$$

— la taille de la fenêtre glissante : ce paramètre est très important puisqu'il a une influence sur la taille et la précision des modèles. Pour observer cette influence, trois valeurs seront considérées : 100, 500 et 1000 ;

— la proportion de sous-échantillons : pour ces tests, cette proportion est fixée à 1, ce qui signifie que tous les éléments de la fenêtre glissante sont utilisés comme centres de noyaux.

Paramétrage de la méthode par GMM L'implémentation des GMM utilisée est la version paramétrique de SmartSifter [Yamanishi 2004]. En plus du nombre k de gaussiennes, deux autres paramètres sont introduits : r, un facteur d'oubli qui permet d'assigner un poids plus faible aux instances plus anciennes dans l'optimisation, et α, un paramètre de stabilité.

Les trois paramètres sont fixés ainsi :

— pour étudier l'influence de k, plusieurs valeurs sont testées : pour les jeux de données à une gaussienne, les valeurs 1 et 5, et pour les jeux de données à huit gaussiennes, les valeurs 4, 8 et 12 ;

— r est équivalent à la taille de la fenêtre pour les KDE : si on considère la suite géométrique des $(1-r)^i$, la somme des éléments tend vers $1/r$, ce qui amène aux valeurs 0.01, 0.002 et 0.001 comme équivalents des valeurs de taille de fenêtre 100, 500 et 1000 ;
— il est recommandé de fixer α entre 1 et 2 [Yamanishi 2004], la valeur utilisée dans les tests est ainsi 1.5.

Evaluation avec une gaussienne Les formes des jeux de données utilisés en dimensions 2 et 3 sont représentés respectivement par les Figures 3.1 et 3.2.

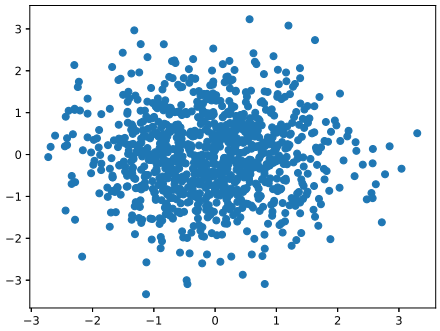

FIGURE 3.1 – Forme du jeu de données à une gaussienne en dimension 2.

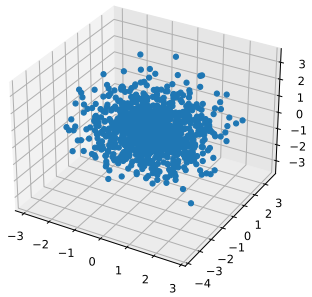

FIGURE 3.2 – Forme du jeu de données à une gaussienne en dimension 3.

Les Figures 3.3, 3.4 et 3.5 montrent respectivement les résultats obtenus en dimensions 2, 3 et 4. On remarque que la méthode par GMM avec $k = 1$ fournit logiquement le classement le plus proche du classement réel et que la distance décroît quand la taille de la fenêtre augmente.

La méthode par KDE a cependant des performances assez proches en comparaison à la distance de référence obtenue pour un classement aléatoire. Elle est également bien plus précise que la méthode par GMM avec $k = 5$ dont la préci-

sion décroît en augmentant la taille de la fenêtre. Cette observation témoigne de l'importance de convenablement fixer le paramètre k pour les GMM.

On peut aussi noter que la précision diminue quand le nombre d'attributs augmente ; en effet, la valeur de référence augmente d'environ 0.6 à 0.9 en passant de 2 à 4 dimensions, et l'augmentation est aussi perceptible sur les autres valeurs.

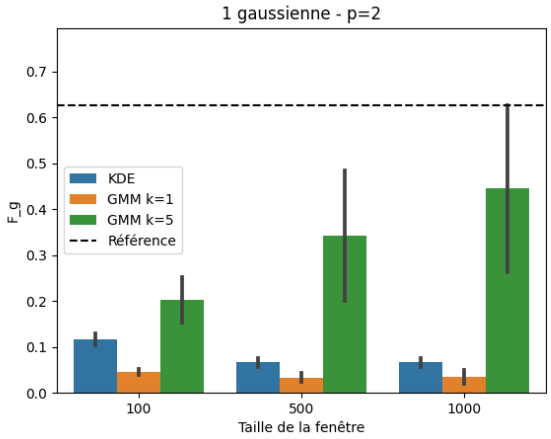

FIGURE 3.3 – Résultats pour les anomalies statistiques – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact sur le jeu à une gaussienne en deux dimensions.

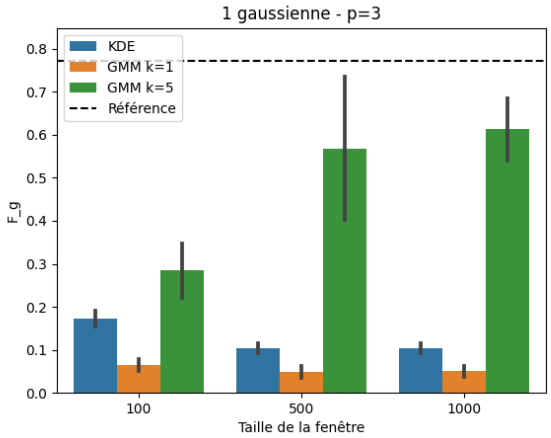

FIGURE 3.4 – Résultats pour les anomalies statistiques – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact sur le jeu à une gaussienne en trois dimensions.

Evaluation avec un mélange de huit gaussiennes Les Figures 3.6 et 3.7 représentent respectivement les jeux de données de huits gaussiennes en dimensions

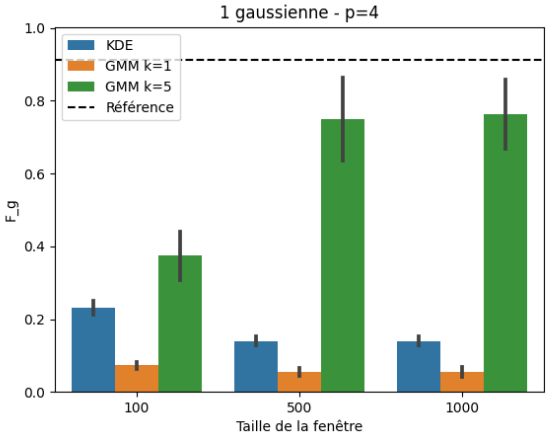

FIGURE 3.5 – Résultats pour les anomalies statistiques – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact sur le jeu à une gaussienne en quatre dimensions.

2 et 3 tandis que les Figures 3.8 et 3.9 présentent les résultats obtenus sur ces jeux de données.

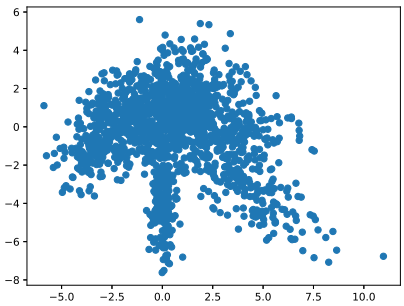

FIGURE 3.6 – Forme du jeu de données à huit gaussiennes en dimension 2.

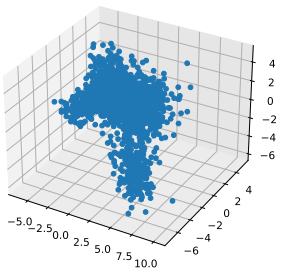

FIGURE 3.7 – Forme du jeu de données à huit gaussiennes en dimension 3.

La précision de la méthode par KDE est cette fois meilleure que celle de la méthode par GMM pour toutes les valeurs de k testées. Cependant, on peut aussi remarquer que celle-ci est plus proche de la valeur de référence obtenue pour des classements aléatoires que ce n'était le cas avec le jeu de données plus simple d'une seule gaussienne. Il est par ailleurs intéressant de noter que le modèle avec $k = 8$, qui devrait être le plus précis, n'obtient les meilleurs résultats parmi les modèles par GMM que dans un cas.

La diminution de la précision des méthodes avec l'augmentation du nombre d'attributs peut ici aussi être constatée. De plus, elle semble décroître plus rapidement pour les méthodes que pour la valeur de référence puisque les valeurs se rapprochent

de la référence en passant de la dimension 2 à la dimension 3.

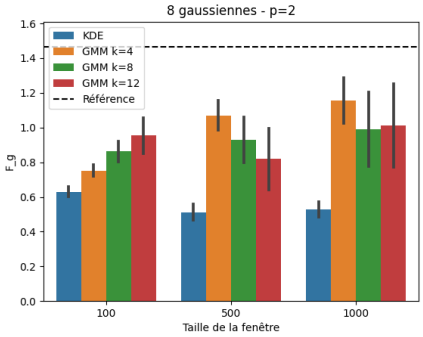

FIGURE 3.8 – Résultats pour les anomalies statistiques – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact sur le jeu à huit gaussiennes en deux dimensions.

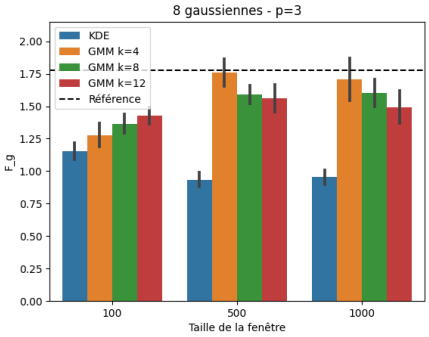

FIGURE 3.9 – Résultats pour les anomalies statistiques – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact sur le jeu à huit gaussiennes en trois dimensions.

3.2.2 Définition de distance

3.2.2.1 Définition de l'anomalie

L'état de l'art, décrit précédemment, définit le voisinage d'une instance comme :

— l'ensemble des instances dans un rayon R selon la définition des (k, R)-anomalies de [Knorr 1998] reprise dans [Kontaki 2011, Palpanas 2003, Subramaniam 2006],

— l'ensemble des k plus proches voisins selon l'approche des kNN [Ramaswamy 2000, Angiulli 2005, Branch 2006, Sheng 2007, Zhang 2007].

Comme annoncé en Section 3.1.2.4, il est possible de mutualiser les modèles en mémoire en détectant les anomalies correspondant à chaque définition à partir de modèles de KDE, ce qui motive le développement de WOLF-KDE. Dans le cas des anomalies de distance, [Palpanas 2003, Subramaniam 2006] proposent d'estimer le nombre de voisins dans un rayon R à partir d'un modèle de KDE. Pour cette raison, la définition du voisinage retenue par la suite est celle provenant des (k, R)-anomalies, s'écrivant précisément :

Définition 6 *Une anomalie de distance est une instance ayant un nombre de voisins dans son R-voisinage inférieur à un seuil.*

Cette définition nécessite de fixer deux éléments : le rayon R de la boule, centrée en l'instance et définissant son voisinage, et un seuil sur le nombre de voisins. Cependant, la valeur du seuil n'influençant pas les résultats selon WOLF-Eval, seul R est à paramétrer pour nos tests.

3.2.2.2 Estimation du nombre de voisins à R

La logique sous-jacente à l'estimation par KDE du nombre de voisins dans un rayon R autour d'une instance provient de l'utilisation d'un Dirac comme fonction noyau pour la KDE, qui donne également la pdf empirique. En effet, soit $\mathcal{B}(\mathbf{p}, R) \subset \mathbb{R}^p$ la boule de rayon R centrée en un point $\mathbf{p} \in \mathbb{R}^p$, on peut retrouver exactement le nombre $n(\mathbf{p}, R) := card(\mathbf{X} \cap \mathcal{B}(\mathbf{p}, R))$ d'éléments de l'ensemble $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l\}$ contenus dans la boule $\mathcal{B}(\mathbf{p}, R)$ par :

$$n(\mathbf{p}, R) = l \int_{\mathcal{B}(\mathbf{p},R))} \tilde{f}_{\mathbf{H},\delta}(u)du\,, \tag{3.9}$$

avec :

$$\tilde{f}_{\mathbf{H},\delta}(\mathbf{x}) = \frac{1}{l} \sum_{\mathbf{x}_i \in \mathbf{X}} \delta(\mathbf{x} - \mathbf{x}_i)\,, \tag{3.10}$$

où δ représente une fonction de Dirac sur $\mathbb{R}^p$, puisque :

$$\int_{\mathcal{B}(\mathbf{p},R))} \delta(u - \mathbf{x})du = \begin{cases} 1 & \text{si} \quad \mathbf{x} \in \mathcal{B}(\mathbf{p}, R))\,, \\ 0 & \text{sinon}\,. \end{cases} \tag{3.11}$$

On peut alors obtenir une approximation de $n(\mathbf{p}, R)$ en remplaçant $\tilde{f_{\mathbf{H},\delta}}(\mathbf{x})$ par une KDE avec une fonction noyau quelconque. En particulier, le noyau de Epanechnikov défini, avec la règle de Scott pour fixer $\mathbf{H}$, dans l'Équation 3.8, est un bon candidat car il est facilement intégrable. Aussi, pour faciliter l'intégration, la boule $\mathcal{B}(\mathbf{p}, R) \subset \mathbb{R}^p$ est approchée par l'hypercube de côté $2R$ centré en $\mathbf{p}$. La Figure 3.10 illustre l'estimation du nombre de voisins en s'appuyant sur le noyau d'Epanechnikov en une dimension.

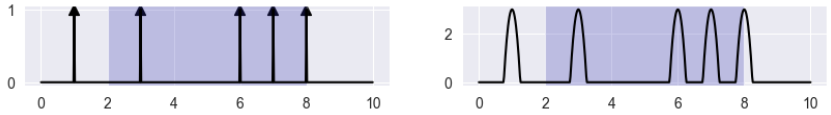

FIGURE 3.10 – Illustration de l'estimation du nombre de voisins à partir d'une KDE dans un intervalle de manière (1, à gauche) exacte en s'appuyant sur des Dirac et (2, à droite) approchée en s'appuyant sur le noyau d'Epanechnikov.

En notant $\mathbf{X}_p$ l'ensemble des éléments de $\mathbf{X}$ tels que la portée du noyau associé se superpose à l'hypercube de côté $2R$ centré en $\mathbf{p}$, c'est-à-dire les $\mathbf{x}_i$ vérifiant, pour tout $1 \leq j \leq p$, $|\mathbf{p}_j - \mathbf{x}_{i,j}| < \mathbf{H}_{j,j} + R$, on obtient, en détaillant sur p dimensions, l'estimation $\tilde{n}(\mathbf{p}, R)$ du nombre de voisins dans $\mathbf{X}$:

$$\tilde{n}(\mathbf{p}, R) = \left(\frac{3}{4}\right)^p \frac{1}{|\mathbf{H}|} \sum_{\mathbf{x}_i \in \mathbf{X}} \prod_{j=1}^{p} \int_{a_j}^{b_j} \left(1 - \left(\frac{\mathbf{x}_j - \mathbf{x}_{i,j}}{\mathbf{H}_{j,j}}\right)^2\right) d\mathbf{x}_j\,, \tag{3.12}$$

avec $a_j = \max\{\mathbf{p} - R, \mathbf{x}_{i,j} - \mathbf{H}_{j,j}\}$ et $b_j = \min\{\mathbf{p} + R, \mathbf{x}_{i,j} + \mathbf{H}_{j,j}\}$ les bornes utiles de l'intégrale.

Or on a :

$$\int_a^b \left(1 - \left(\frac{\mathbf{x}_j - \mathbf{c}}{H}\right)^2\right) d\mathbf{x}_j = \left[\mathbf{x}_j - \frac{(\mathbf{x}_j - \mathbf{c})^3}{3H^2}\right]_{\mathbf{x}_j=a}^{\mathbf{x}_j=b} = (b-a) - \frac{1}{3H^2}((b-\mathbf{c})^3 - (a-\mathbf{c})^3)\,. \tag{3.13}$$

Après intégration de (3.12) et d'après (3.13), on obtient finalement :

$$\tilde{n}(\mathbf{p}, R) = \left(\frac{3}{4}\right)^p \frac{1}{|\mathbf{H}|} \sum_{\mathbf{x}_i \in \mathbf{X_p}} \prod_{j=1}^{p} \Big((b_j - a_j) - \frac{1}{3\mathbf{H}_{j,j}^2}((b_j - \mathbf{x}_{i,j})^3 - (a_j - \mathbf{x}_{i,j})^3)\Big)\,. \tag{3.14}$$

La grandeur peut ainsi être calculée uniquement à partir de la largeur de bande $\mathbf{H}$ et des éléments de $\mathbf{X}_p$.

3.2.2.3 Calcul exact dans une fenêtre glissante

Dans [Kontaki 2011], plusieurs approches sont proposées pour calculer, de manière exacte, le nombre de voisins dans un rayon R à travers une requête dans une fenêtre glissante du flux de données. Pour faciliter cette requête, la fenêtre glissante est stockée dans dans une structure appelée "M-tree" [Ciaccia 2001].

Un M-tree est un arbre où chaque noeud est associé à la plus petite hyper-sphère contenant le sous-arbre dont il est la racine. Les noeuds ne peuvent contenir qu'un nombre fixe d'éléments ; ces éléments sont des instances dans le cas de feuilles de l'arbre et d'autres noeuds dans le cas de noeuds non-feuilles. Les M-tree permettent de retrouver le nombre d'éléments à une distance R d'un point $\mathbf{p}$ en évitant de parcourir tout l'arbre ; en effet, soit $\mathbf{c}$ le centre de l'hyper-sphère associée à un noeud et $R_\mathbf{c}$ son rayon, si $d(\mathbf{p}, \mathbf{c}) > R + R_\mathbf{c}$, avec $d(\mathbf{p}, \mathbf{c})$ la distance entre $\mathbf{p}$ et $\mathbf{c}$, alors le noeud ne contient aucun voisin de $\mathbf{p}$. De plus, en utilisant l'inégalité triangulaire, on peut éviter d'avoir à calculer toutes les distances lors d'une requête en gardant en mémoire les distances entre les centres des hyper-sphères des noeuds parents et enfants [Ciaccia 2001].

La partie la plus complexe dans l'utilisation de M-tree avec des flux de données est le maintien de la structure à travers les ajouts et suppressions successifs d'instances. Pour la suppression d'une instance, on peut simplement la retirer de la liste des enfants de son noeud parent et mettre à jour les rayons d'hyper-sphères qui sont impactés en remontant dans l'arbre. Si le noeud parent ne contient plus d'éléments, on peut de la même manière le supprimer. Pour l'ajout de nouvelles instances, il faut dans un premier temps rechercher le noeud feuille dont le centre est le plus proche. La mise à jour des rayons peut alors être remontée comme pour la suppression d'une instance. La difficulté survient lorsqu'un noeud dépasse le nombre maximal d'enfants, auquel cas il faut suivre une procédure de division particulière

et élire de nouvelles instances comme centres d'hyper-sphères pour les noeuds créés [Ciaccia 2001].

Le maintien d'une structure augmente le nombre de paramètres à conserver en mémoire, et bien que le coût algorithmique de la requête du nombre de voisins soit faible, le coût de la mise à jour de la structure dans une fenêtre glissante n'est pas négligeable. En revanche, cette approche permet de trouver exactement le nombre de voisins.

3.2.2.4 Évaluation de l'estimation

Description de l'expérimentation Pour les anomalies de distance, le calcul du classement correct nécessaire à WOLF-Eval peut être fait sur n'importe quel jeu de données en prenant la matrice des distances entre toutes les instances. Finalement, le score d'une instance $\mathbf{x}_i$ est donné par $w_i = 1/(1 + n(\mathbf{x}_i, R))$.

Le jeu de données utilisé ici est un jeu en trois dimensions constitué d'un cluster normal contenant 1000 instances, générées par une distribution gaussienne centrée réduite, et de 20 instances anormales environnantes, générées par une distribution uniforme. La Figure 3.11 donne la forme d'un tel jeu de données.

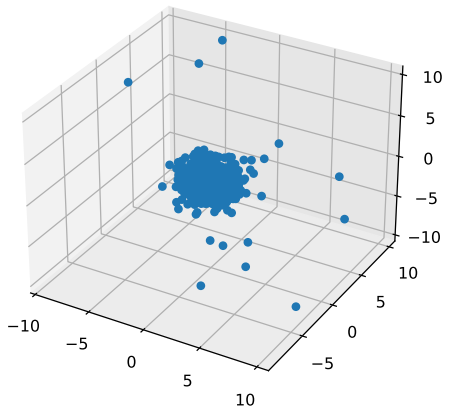

FIGURE 3.11 – Forme du jeu de données utilisé pour l'évaluation des méthodes dans le cadre des évaluations d'anomalies de distance et de métrique locale.

L'influence de deux paramètres sera étudiée lors des expérimentations :

— le paramètre R définissant le voisinage : les valeurs testées sont 0.5, 1 et 2 ;
— la taille de la fenêtre : les valeurs testées sont 50, 100, 500 et le jeu de données complet. Dans tous les cas, la moitié du jeu de données est utilisé pour la partie évaluation afin de générer le classement ; dans le cas du jeu de données complet, cela signifie que les instances utilisées pour l'évaluation ont déjà été intégrées au modèle, ce qui n'est pas le cas pour les autres valeurs

testées où les instances sont évaluées avant d'être intégrées, comme dans un apprentissage en ligne.

En plus de la distance au classement exact, la durée d'évaluation des instances est mesurée, en secondes, pour étudier le coût calculatoire des méthodes comparées. Pour obtenir les résultats fournis, les tests sont réalisés en PYTHON (3.10) avec un processeur 11th Gen Intel® Core™ i7-11800H @ 2.30GHz et 32Go de RAM.

Paramétrage des méthodes Pour la méthode approchée par KDE, on a déjà fixé le noyau d'Epanechnikov et la règle de Scott. Les seuls autres paramètres à fixer sont le paramètre du voisinage R et la taille de la fenêtre glissante, dont on a déjà précisé les valeurs. Comme pour le cas des anomalies statistiques, la proportion de sous-échantillons est fixée à 1 ; on prend donc toutes les instances de la fenêtre comme centres de noyaux.

La méthode exacte par M-tree ne nécessite que de fixer les variables de l'expérience ainsi qu'un troisième paramètre M qui correspond au nombre maximal d'enfants pour chaque noeud et que l'on fixe à $M = 5$. Ce paramètre n'affecte que la taille du modèle et la vitesse de traitement, pas la précision des résultats.

Restitution des résultats Les résultats en précision obtenus pour les différentes valeurs de R sont fournis dans les Figures 3.12, 3.14 et 3.16 tandis que les résultats en temps de traitement sont fournis dans les Figures 3.13, 3.15 et 3.17.

Il est intéressant de noter que la précision du résultat, pour ce jeu de données assez simple, ne dépend que très peu de la taille de la fenêtre dans le cas de la méthode approchée par KDE. Aussi, le classement est presque toujours meilleur que celui renvoyé par la méthode exacte. La différence diminue cependant lorsque R et la taille de la fenêtre augmentent. Enfin, lorsque l'entièreté du jeu de données est intégré au M-tree, l'approche exacte atteint logiquement une distance F_g nulle dans la majorité des cas tandis qu'il n'y a pas d'améliorations perceptibles pour la méthode approchée.

En revanche, la durée d'évaluation est toujours plus importante pour la méthode par KDE à cause de la recherche des centres de noyaux se superposant sur le rayon de requête des voisins. La complexité augmentant avec la taille de la fenêtre glissante, il semble préférable de choisir des tailles limitées puisque cela n'impacte que peu la précision des résultats.

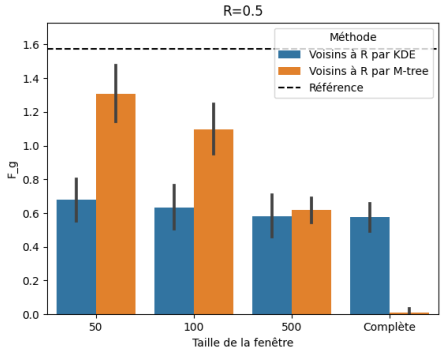

FIGURE 3.12 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 0.5$.

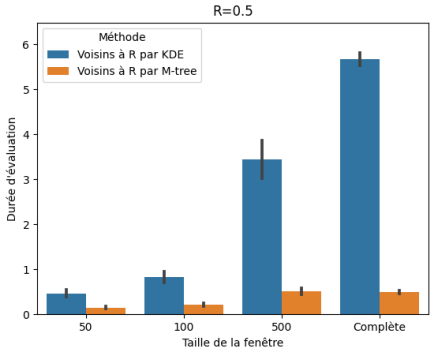

FIGURE 3.13 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 0.5$.

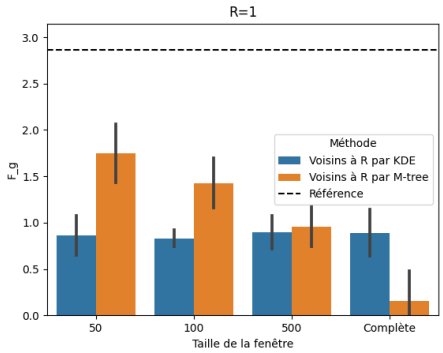

FIGURE 3.14 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 1$.

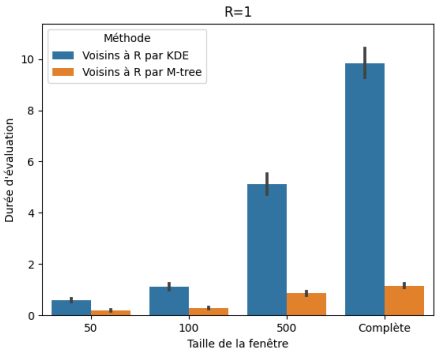

FIGURE 3.15 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 1$.

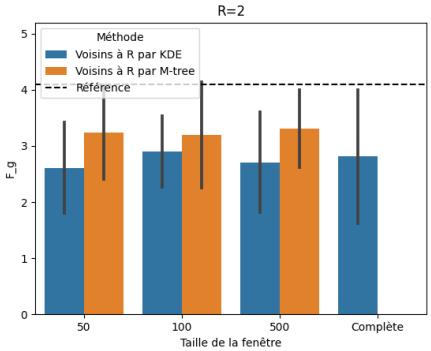

FIGURE 3.16 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 2$.

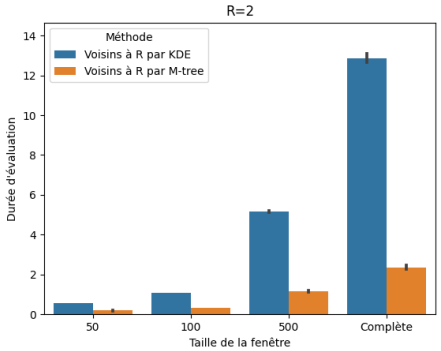

FIGURE 3.17 – Résultats pour les anomalies de distance – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 2$.

3.2.3 Définition de densité locale

3.2.3.1 Définition de l'anomalie

La méta-définition pour les anomalies de densité locale est :

Définition 7 *Une anomalie de densité locale est une instance dont la métrique de densité locale associée est supérieure à un seuil.*

La définition nécessite encore de fixer deux éléments : une métrique de densité locale associée à chaque instance et un seuil.

Dans l'état de l'art proposé, deux métriques ont été présentées pour évaluer le degré d'anormalité locale d'une instance : le LOF [Breunig 2000] et le MDEF [Papadimitriou 2003]. Bien que plusieurs approches aient cherché à coupler les avantages du LOF et des KDE [Gao 2011, Schubert 2014, Tang 2017], celles-ci reposent tout de même sur l'utilisation des kNN, dont la recherche peut être coûteuse. A notre connaissance, le MDEF est donc le meilleur candidat pour une estimation par KDE au sein de WOLF-KDE. Cette estimation est notamment proposée par [Subramaniam 2006] en suivant la description d'un calcul approché du MDEF fournie par [Papadimitriou 2003].

3.2.3.2 Calcul exact du MDEF

Contrairement au LOF, le MDEF peut être calculé uniquement à partir de requêtes de nombres de voisins à une certaine distance. Précisément, deux distances sont considérées, d'où le terme "multi-granularité" dans le sigle MDEF : R et $r = 2^{-\alpha}R$ avec $\alpha \in \mathbb{N}^*$.

Alors que le LOF compare la densité d'accessibilité locale d'une instance à celle de ses kNN, le MDEF compare plus simplement le nombre de voisins à r d'une

instance à celui moyen dans son voisinage à R [Papadimitriou 2003]. En notant $n(\mathbf{p}, r)$ le nombre d'instances à une distance r de $\mathbf{p}$ et $N(\mathbf{p}, R)$ l'ensemble des instances à une distance R de $\mathbf{p}$, on a :

$$\tilde{n}(\mathbf{p}, r, R) = \frac{1}{card(N(\mathbf{p}, R))} \sum_{\mathbf{x} \in N(\mathbf{p}, R)} n(\mathbf{x}, r) \tag{3.15}$$

et :

$$\mathbf{MDEF}_{r,R}(\mathbf{p}) = 1 - \frac{n(\mathbf{p}, r)}{\tilde{n}(\mathbf{p}, r, R)} \,. \tag{3.16}$$

Aussi, contrairement au LOF, [Papadimitriou 2003] propose un seuil dynamique sur le MDEF, permettant de définir le seuil de la Définition 7, et une instance $\mathbf{p}$ est considérée comme anormale si :

$$\mathbf{MDEF}_{r,R}(\mathbf{p}) > k_{\tilde{\sigma}} \tilde{\sigma}_{\mathbf{MDEF}}(\mathbf{p}, r, R) \tag{3.17}$$

où

$$\tilde{\sigma}_{\mathbf{MDEF}}(\mathbf{p}, r, R) = \frac{\sigma(\mathbf{p}, r, R)}{\tilde{n}(\mathbf{p}, r, R)} \tag{3.18}$$

avec $\sigma(\mathbf{p}, r, R)$ l'écart-type des $n(\mathbf{x}, r)$ pour $\mathbf{x} \in N(\mathbf{p}, R)$ et $k_{\tilde{\sigma}}$ un facteur, généralement fixé à 3, tel que la probabilité qu'une instance soit considéré anormale est bornée par :

$$P\big(\mathbf{MDEF}_{r,R}(\mathbf{p}) > k_{\tilde{\sigma}} \tilde{\sigma}_{\mathbf{MDEF}}(\mathbf{p}, r, R)\big) \leq \frac{1}{k_{\tilde{\sigma}}^2} \,. \tag{3.19}$$

La métrique de densité locale utilisée est donc précisément le MDEF corrigé $\mathbf{MDEF}_{r,R}(\mathbf{p}) - k_{\tilde{\sigma}} \tilde{\sigma}_{\mathbf{MDEF}}(\mathbf{p}, r, R)$ avec un seuil à 0 sur ce score, et on obtient précisément la définition pour les anomalies de densité locale dans cette étude :

Définition 8 *Une anomalie de densité locale est une instance pour laquelle le MDEF corrigé est supérieur à* 0*.*

Aussi, puisqu'il ne nécessite que des requêtes de voisinages, le calcul exact du MDEF corrigé peut, comme dans le cas des anomalies de distance, être réalisé à l'aide d'un M-tree.

3.2.3.3 Calcul approché du MDEF

L'article de [Papadimitriou 2003] propose également un moyen d'estimer certaines des grandeurs nécessaires au calcul du MDEF, à savoir $\tilde{n}(\mathbf{p}, r, R)$ et $\sigma(\mathbf{p}, r, R)$, afin d'éviter de rechercher exactement l'ensemble $N(\mathbf{p}, R)$ et le nombre de voisins de chacun de ses éléments.

Pour ce faire, l'utilisation de grilles en p dimensions est proposée pour représenter le voisinage à R de $\mathbf{p}$ en s'appuyant sur le fait que $r = 2^{-\alpha} R$ pour diviser cette grille en cellules. Ainsi, la grille complète, un hyper-cube centré en $\mathbf{p}$ et de côté $2R$, est divisée en $2^{\alpha p}$ cellulles, des hyper-cubes de côtés $2r$ représentant des r-voisinages dans le R-voisinage. Le nombre d'instances c_i dans chaque cellule $\mathcal{C}_i$ de la grille $\mathcal{G}(\mathbf{p}, r, R)$, $1 \leq i \leq 2^{\alpha p}$, est alors dénombré.

En posant $S_q(\mathbf{p}, r, R) = \sum_{i=1}^{2^{\alpha p}} c_i^q$, les estimations proposées sont :

$$\tilde{n}(\mathbf{p}, r, R) \approx \frac{S_2(\mathbf{p}, r, R)}{S_1(\mathbf{p}, r, R)} \tag{3.20}$$

et :

$$\sigma(\mathbf{p}, r, R) \approx \sqrt{\frac{S_3(\mathbf{p}, r, R)}{S_1(\mathbf{p}, r, R)} - \frac{S_2^2(\mathbf{p}, r, R)}{S_1^2(\mathbf{p}, r, R)}}\,. \tag{3.21}$$

Le MDEF peut ainsi être approché uniquement en estimant le nombre d'instances dans des hyper-cubes, ce qui peut être estimé par KDE comme proposé par [Subramaniam 2006] et démontré précédemment dans le cadre des anomalies de distance.

Notons que le nombre de cellules, et donc de paramètres utilisés pour le calcul approché du MDEF corrigé, est exponentiel selon α et p. On a déjà fixé $2 \leq p \leq 4$, mais le paramètre α doit également rester faible.

3.2.3.4 Évaluation de l'estimation

Description de l'expérimentation De manière similaire aux anomalies de distance, la mise en place de WOLF-Eval pour les anomalies de densité locale peut être réalisée sur n'importe quel jeu de données en prenant la matrice des distances entre les instances. Aussi, le jeu de données utilisé est le même que celui utilisé pour les anomalies de distance et représenté par la Figure 3.11. Enfin, la durée de la phase d'évaluation des instances est à nouveau mesurée et restituée.

Comme décrit précédemment, la densité locale retenue est le MDEF corrigé. Le score utilisé pour une instance $\mathbf{x}_i$ est donc $w_i = \mathbf{MDEF}_{r,R}(\mathbf{x}_i) - k_{\tilde{\sigma}} \tilde{\sigma}_{\mathbf{MDEF}}(\mathbf{x}_i, r, R)$. Deux méthodes sont comparées pour son estimation dans des fenêtres glissantes : le calcul approché par KDE et le calcul exact par M-tree. Pour le classement correct et le classement fourni par l'utilisation d'un M-tree, le MDEF corrigé est calculé exactement. Pour le classement fourni par l'utilisation d'une KDE, le MDEF est approché.

Les variables expérimentales sont les mêmes que dans le cas des anomalies de distance, à savoir la taille de la fenêtre et R. La première prend les mêmes valeurs que précédemment tandis que la seconde prend les valeurs 1, 1.5 et 2.

Paramétrage des méthodes En dehors des variables expérimentales, les autres paramètres propres au MDEF sont fixés à $k_{\tilde{\sigma}} = 3$, comme préconisé dans l'article décrivant le MDEF [Papadimitriou 2003], et $\alpha = 2$ pour réduire la complexité en limitant le nombre de cellules. Les paramètres des KDE et du M-tree sont quant à eux fixés comme pour le cas des anomalies de distance.

Restitution des résultats Les résultats en précision obtenus pour les différentes valeurs de R sont fournis dans les Figures 3.18, 3.20 et 3.22 tandis que les résultats en temps de traitement sont fournis dans les Figures 3.19, 3.21 et 3.23.

Contrairement au cas des anomalies de distance, l'écart de précision entre la méthode approchée par KDE et la méthode exacte par M-tree est faible peu importe la taille de la fenêtre. Cependant, la précision évolue toujours peu pour la méthode par KDE tandis qu'elle augmente avec la taille de la fenêtre pour la méthode par M-tree. Pour les deux méthodes, la précision augmente également avec R, à l'opposé de ce qui pouvait être observé pour les anomalies de distance.

La durée d'évaluation suit une évolution similaire au cas des anomalies de distance à la différence qu'on peut percevoir une croissance notable avec l'augmentation de R pour la méthode par M-tree ; pour $R = 2$ et une fenêtre de 500 instances, le temps de traitement rattrape même la méthode par KDE.

Il est tout de même important de noter que le temps de traitement le plus long reste suffisamment court pour traiter plusieurs instances par seconde.

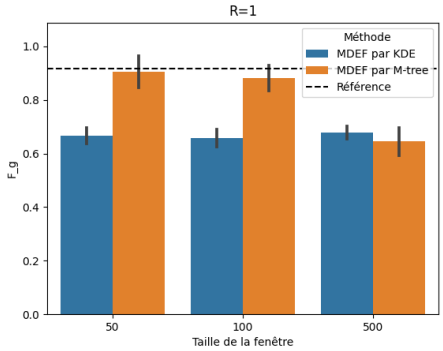

FIGURE 3.18 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 1$.

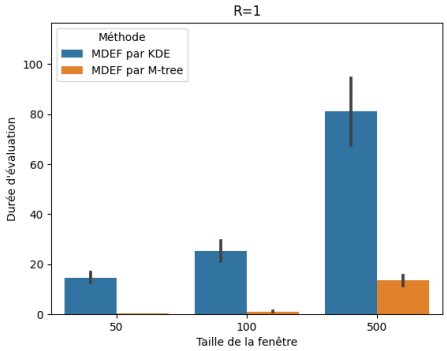

FIGURE 3.19 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 1$.

3.3 Récapitulatif et discussion

Pour conclure ce chapitre, on dresse le bilan des travaux réalisés pour concevoir le cadre opérationnel appliqué WOLF-KDE et l'évaluer avec la méthode WOLF-Eval. Une discussion est ensuite proposée concernant les possibilités offertes par un tel cadre.

3.3.1 Récapitulatif de la phase expérimentale

Globalement, pour les trois définitions proposées et étudiées dans ce chapitre, les méthodes à base de KDE, en plus de permettre de ne maintenir qu'un unique modèle pour traiter toutes les définitions d'anomalies, obtiennent de meilleures performances que les méthodes auxquelles elles sont comparées.

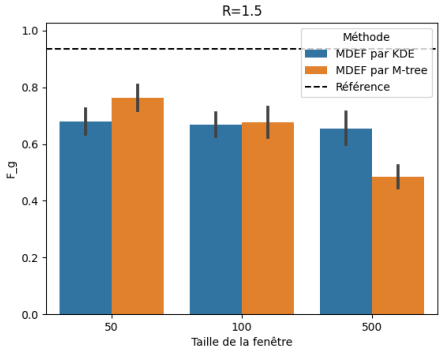

FIGURE 3.20 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 1.5$.

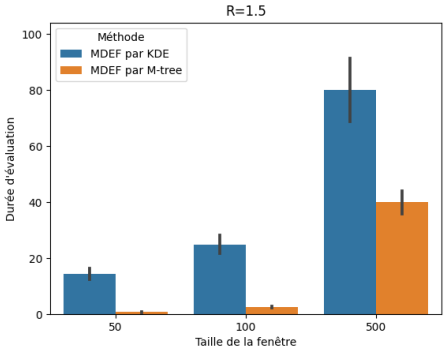

FIGURE 3.21 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 1.5$.

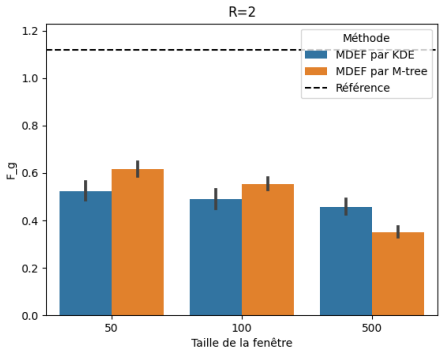

FIGURE 3.22 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la distance (F_g) au classement exact avec $R = 2$.

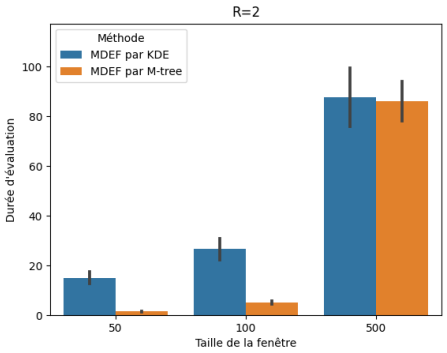

FIGURE 3.23 – Résultats pour les anomalies de densité locale – Représentation de la moyenne et de l'écart-type de la durée d'évaluation de 510 instances en secondes avec $R = 2$.

Dans le cas des anomalies statistiques, la méthode par GMM est plus précise pour traiter des mélanges gaussiens dégénérés, c'est-à-dire avec une seule gaussienne, mais elle est en retrait pour un mélange de huit gaussiennes. De plus, les performances des méthodes par KDE s'améliorent avec la taille de la fenêtre, ce qui n'est pas toujours le cas pour les méthodes par GMM. Ces dernières sont également sujettes au choix du paramètre k, difficile à fixer dans des cas d'application réels.

Pour les anomalies de distance et de métrique locale, il est surprenant de voir qu'on obtient une meilleure précision avec les méthodes approchées par KDE qu'avec les méthodes exactes par M-tree lorsqu'on prend des fenêtres glissantes. Assez logiquement, plus la taille de la fenêtre est grande et plus les méthodes exactes sont précises, et ce jusqu'à donner exactement le classement attendu lorsque tout le jeu de données est pris en compte. Cependant, la précision des méthodes par KDE varie, quant à elle, assez peu avec l'évolution de la taille de la fenêtre, ce qui permet leur utilisation à des coûts acceptables en occupation mémoire et temps de traitement.

Enfin, la méthode par KDE pour les anomalies basées densité locale a l'avantage de proposer un seuil pré-défini, facilitant grandement son paramétrage.

3.3.2 Discussions

Ce chapitre propose le développement d'un cadre opérationnel pour la détection d'anomalies dans les WSNs sous la forme d'une boîte à outils. Celle-ci est divisée en compartiments associés à une définition de l'anomalie et contenant un ensemble de méthodes, répondant aux spécificités des WSNs et des flux de données, qui permettent de détecter les anomalies suivant les différentes définitions.

L'intérêt principal du cadre proposé, nommé WOLF, est de permettre une évaluation non supervisée des méthodes et également de faciliter leur sélection par un expert. De plus, le fait que chaque élément de la boîte à outil soit associé à une définition, donc à une sémantique particulière, permet de donner un sens aux anomalies détectées.

Cependant, multiplier les compartiments dans la boîte à outils pourrait avoir l'inconvénient de multiplier les paramètres en mémoires dans les noeuds du WSN si chaque méthode était associée à un modèle différent. Pour cette raison, nous avons proposé le cadre unifié WOLF-KDE dont les méthodes reposent à chaque fois sur les modèles de KDE avec la fonction d'Epanechnikov comme noyau. Les modèles peuvent ainsi être mutualisés, réduisant grandement les paramètres à conserver en mémoire.

L'utilisation des KDE a tout de même certains inconvénients :

— le choix de la largeur de bande est un paramètre crucial et pouvant avoir des conséquences importantes sur la précision des modèles ; en pratique, des largeurs de bandes dynamiques sont souvent préférées pour s'adapter à la densité locale des données [Ristic 2008, Laxhammar 2009, Schubert 2014],

— le choix de la fonction noyau est également important ; le noyau d'Epanechnikov a été utilisé dans ces tests car il est facile à intégrer et fréquemment

utilisé, mais d'autres noyaux auraient pu donner des résultats similaires voire meilleurs, et bien qu'on ait conservé le même noyau pour chaque définition, en pratique le modèle n'est constitué que des centres de noyaux et de l'écart-type pour le calcul de la largeur de bande, donc l'utilisation de fonctions noyau différentes est possible,

— en particulier dans le cas des anomalies statistiques, la précision est dépendante du nombre d'échantillons utilisés pour l'estimation, or le coût en occupation mémoire et en temps de traitement des méthodes par KDE augmente rapidement en multipliant le nombre d'échantillons, surtout pour les anomalies de métrique locale.

Dans les chapitres suivants, la mise en commun des différentes méthodes est étudiée et une alternative à l'approche par KDE est proposée.

Chapitre 4

Approche intégrée de détection d'anomalies

Le chapitre précédent a présenté WOLF, un cadre opérationnel prenant la forme d'une boîte à outils dans laquelle plusieurs définitions de l'anomalie peuvent être traitées à travers différentes méthodes de détection d'anomalies. Cependant, en suivant la même motivation que pour les méthodes ensemblistes, mentionnées en Section 2.1.2, il serait intéressant d'intégrer conjointement les différentes méthodes au sein de WOLF, et en particulier dans WOLF-KDE, en combinant leurs résultats. Ce quatrième chapitre présente d'abord une approche hors ligne, SuMeRI, développée dans [Ducharlet 2020], avant de proposer une adaptation aux flux de données. Cette adaptation est finalement comparée à l'approche consistant à appliquer les méthodes de manière indépendante.

Sommaire

4.1 SuMeRI pour la combinaison de méthodes

Cette première section présente l'approche SuMeRI (Successive Methods Run Iteratively), proposée pour détecter différents types d'anomalies *dans un contexte hors ligne* tout en facilitant le paramétrage des méthodes. Les limites de SuMeRI sont également développées.

4.1.1 Motivations

Les travaux décrits dans [Ducharlet 2020] sont réalisés dans un contexte hors ligne ; on suppose que la distribution des données n'évolue pas dans le temps ou qu'un ré-apprentissage régulier des modèles est suffisant pour répondre à cette évolution. On commence ici par traiter du problème de manière hors ligne avant de chercher une adaptation en ligne pour traiter des flux de données issus des réseaux de capteurs.

L'approche proposée est motivée par deux constats :

— il n'existe pas de solution universelle au problème de détection d'anomalies ; une seule méthode ne peut pas détecter tous les types d'anomalies,

— il est difficile de paramétrer les méthodes dans un contexte non supervisé.

Le premier constat est traité dans SuMeRI par l'application successive de différentes méthodes, chacune ayant pour vocation de traiter un type d'anomalies différent. Le second constat est traité par un apprentissage itératif du modèle, avec un paramétrage générique des méthodes, en modifiant à chaque itération le jeu d'entraînement. Cet apprentissage itératif est inspiré par l'approche dite "diagnostic", mentionnée dans [Hodge 2004], qui consiste à élaguer itérativement les anomalies et adapter le modèle jusqu'à ce qu'il n'y ait plus d'anomalies détectées.

4.1.2 Phase itérative

Pour chaque méthode de détection d'anomalies, on souhaite disposer d'un paramétrage suffisamment générique pour pouvoir estimer les "anomalies prédominantes" dans le jeu de données. Il s'agit d'anomalies suffisamment évidentes pour qu'on puisse supposer qu'elles seraient détectées par un grand nombre de paramétrages différents de la méthode. L'objectif est cependant de pouvoir détecter des anomalies moins évidentes tout en maintenant le même paramétrage générique. Pour ce faire, les anomalies principales précédemment identifiées sont retirées du jeu de données d'entraînement et un modèle est de nouveau appris, identifiant de nouvelles anomalies principales qui étaient masquées jusqu'alors par les précédentes. L'intuition derrière cette approche est qu'il sera possible, à force d'itérations, d'élaguer les anomalies par strates jusqu'à atteindre le *noyau normal* du jeu de données.

Idéalement, il serait souhaitable que le nombre d'anomalies principales détectées au fil des itérations décroisse et atteigne zéro lorsqu'on atteint le noyau normal. Cependant, il est difficile de vérifier cette convergence tout en acceptant n'importe quelle méthode dans SuMeRI. Aussi, un critère d'arrêt générique a été défini dans

l'article [Ducharlet 2020]. Le processus de la phase itérative de SuMeRI est décrit par la Figure 4.1.

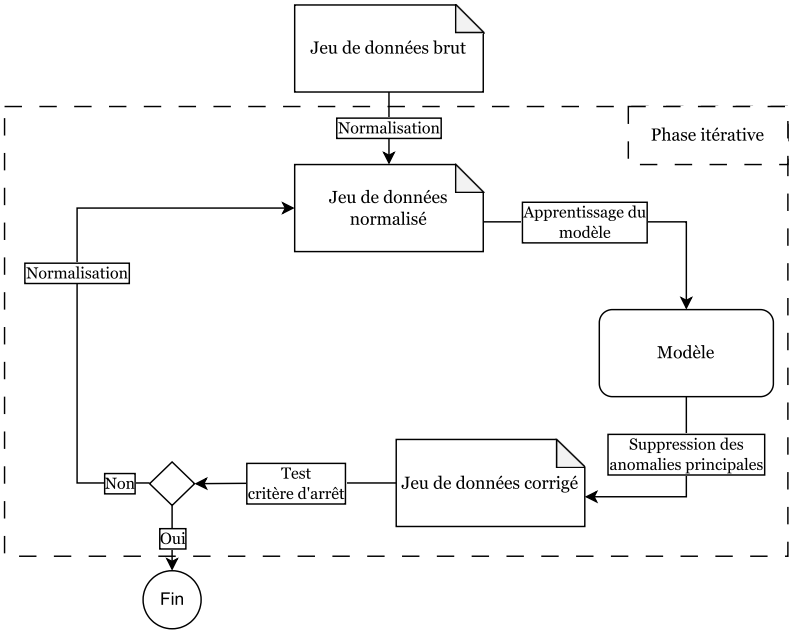

FIGURE 4.1 – Schéma représentant la phase itérative de SuMeRI pour une méthode donnée.

4.1.2.1 Suppression des anomalies principales

La définition des anomalies principales s'appuie sur l'écart entre les scores d'anomalies dans le jeu de données. L'intuition derrière cette approche est que les instances normales ont un score d'anomalie similaire tandis que le score d'anomalie des instances anormales s'écarte de plus en plus.

Une fois le modèle "appris", le score s_i de chaque instance, indexée par i, est calculé. Ces scores sont ensuite triés par ordre croissant pour obtenir la suite des $(s_i)_{1\leq i\leq n}$, où n est le nombre d'instances, allant du score le plus normal au score le plus anormal. Soit t_1 le seuil sur le score permettant au modèle de séparer l'ensemble des scores des instances considérées comme normales S_{in} de l'ensemble des scores des instances considérées comme anormales S_{out}, on définit un second seuil t_2 comme le score correspondant au plus grand écart entre deux scores successifs et différents dans S_{out} :

$$t_2 = \underset{s_i \in S_{out}}{\arg\max}(\min_{s_j \neq s_i}(|s_j - s_i|)) \,. \tag{4.1}$$

Le seuil t_2 est utilisé pour séparer les instances simplement considérées comme anormales des anomalies principales dans S_{out}. Ainsi, les instances correspondant à un score $s_i > t_2$ sont retirées du jeu de données pour préparer l'itération suivante.

4.1.2.2 Critère d'arrêt

En définissant

$$d_{max}(S) = \max_{s_i \in S}(\min_{s_j \neq s_i}(|s_j - s_i|)) \tag{4.2}$$

comme la séparation maximale entre deux scores successifs et différents dans un ensemble de scores trié S, le critère d'arrêt proposé est :

$$d_{max}(S_{in}) \geq d_{max}(S_{out}) \tag{4.3}$$

qui correspond à l'instant où le plus grand écart entre deux scores successifs ne se trouve plus dans l'ensemble S_{out} mais dans l'ensemble S_{in}. En suivant l'intuition proposée pour définir les anomalies principales, on suppose qu'une fois toutes les anomalies retirées, le jeu de données est suffisamment homogène pour que le plus grand écart entre les scores successifs puisse aussi bien se trouver dans S_{in} que dans S_{out}. Aussi, S_{in} comprenant bien plus d'éléments que S_{out}, il est alors plus probable qu'il se retrouve dans S_{in}.

4.1.3 Phase séquentielle

En sortie d'une phase d'apprentissage itératif, on retrouve le modèle appris ainsi que le jeu de données corrigé à travers les itérations. Ce jeu de données est utilisé en entrée de l'apprentissage itératif d'une nouvelle méthode, définissant ainsi la phase séquentielle de SuMeRI. Contrairement à un apprentissage parallèle où la phase itérative serait appliquée indépendamment sur chaque méthode, les anomalies principales retirées lors de l'apprentissage d'une méthode ne sont pas présentes dans le jeu de données d'apprentissage de la méthode suivante.

L'avantage de cette approche est qu'elle permet l'utilisation de méthodes dont la précision des modèles est sensible à la présence d'anomalies dans le jeu d'entraînement. Aussi, il est préférable d'utiliser des méthodes robustes dans les premières phases de la séquence de SuMeRI ; les méthodes plus sensibles, dont les résultats seraient perturbés par la présence d'anomalies, étant réservées pour les dernières phases pour lesquelles les anomalies principales ont déjà été retirées du jeu d'entraînement.

La Figure 4.2 représente le processus complet de SuMeRI. Durant l'apprentissage, les méthodes sont appliquées itérativement une par une et les modèles sont générés ainsi. Le jeu de données corrigé en sortie d'une phase itérative est utilisé en entrée de la suivante. Lors de l'application des modèles, de nouvelles instances sont évaluées par les différents modèles obtenus, et on obtient ainsi un total de m labels par instance, correspondant aux m méthodes choisies.

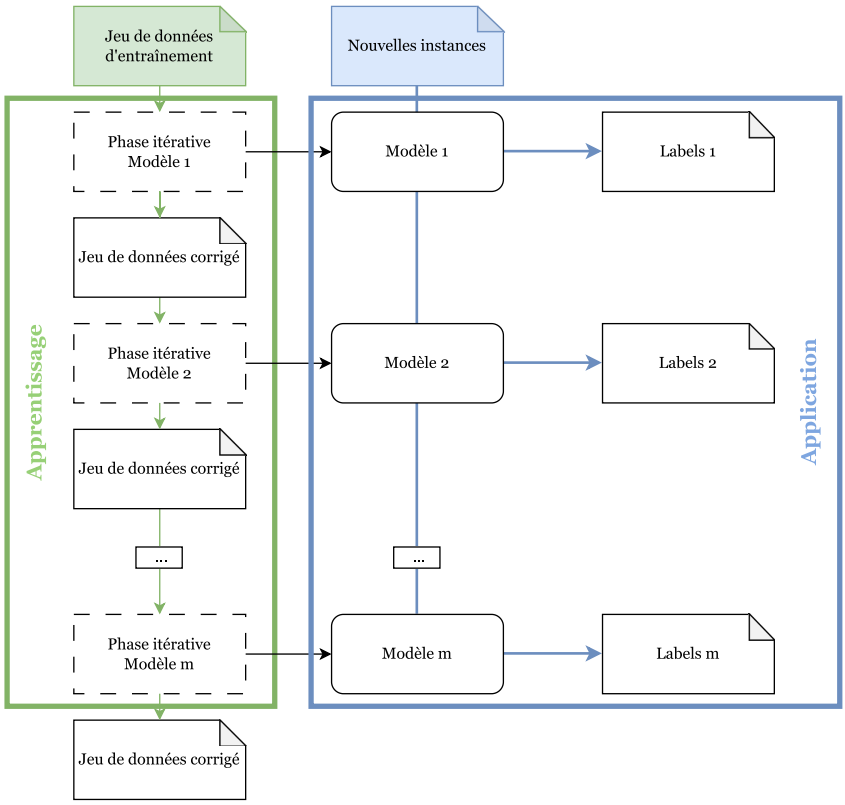

FIGURE 4.2 – Schéma représentant le processus complet de SuMeRI (apprentissage en vert et application en bleu).

4.1.4 Limites identifiées

SuMeRI propose une solution générique au problème de détection d'anomalies en permettant de combiner différentes méthodes, et ainsi de détecter différents types d'anomalies. Cette solution souffre tout de même de problèmes liés aux conditions fortes qu'elle suppose pour les méthodes employées :

— pour chaque méthode, il est nécessaire de fournir un paramétrage "générique" ; trouver ce paramétrage, que l'on a supposé permettre la détection d'anomalies principales, nécessite un grand nombre de tests, et il est possible qu'il n'existe simplement pas,

— le fonctionnement de la phase itérative se base sur deux hypothèses fortes concernant le score renvoyé par les modèles appris ; (1) plus les instances sont anormales, plus l'écart entre les scores successifs s'écarte et (2) en retirant les anomalies principales, ces écarts s'homogénéisent pour atteindre la condition d'arrêt,

— retirer les anomalies du jeu d'entraînement dans l'apprentissage séquentiel suppose que les méthodes employées ne créent pas de faux positifs, sinon il est possible de perdre une partie de la connaissance en supprimant des instances normales.

Enfin, comme ce sera décrit dans la section suivante, SuMeRI n'est pas applicable en l'état dans le contexte des WSNs.

4.2 Une adaptation pour des flux de données

L'objectif de cette section est de reprendre la logique de SuMeRI et de l'adapter aux WSNs, et en particulier au cadre opérationnel WOLF, proposé dans le Chapitre 3. On cherche donc à passer d'une application hors ligne à une application en ligne.

4.2.1 De l'itératif à l'incrémental

L'apprentissage itératif de SuMeRI est coûteux et ne peut pas être réalisé en continu. Si la distribution des données change, il est nécessaire de ré-exécuter la phase d'apprentissage. De plus, l'aspect itératif propose de retirer les anomalies principales, couche par couche, au fil des itérations et de la succession des méthodes. Mais, dans un flux de données, la notion d'anomalie principale n'a de sens qu'à un instant donné et l'approche itérative parait inadaptée.

Pour ces raisons, l'adaptation de SuMeRI proposée au sein de WOLF remplace l'aspect itératif par l'aspect incrémental de l'apprentissage en ligne. Plus spécifiquement, l'apprentissage itératif *en amont* de la détection est assez logiquement remplacé par un apprentissage en continu *pendant* la détection. La Figure 4.3 correspond à la simplification de la Figure 4.1 en passant de l'itératif à l'incrémental.

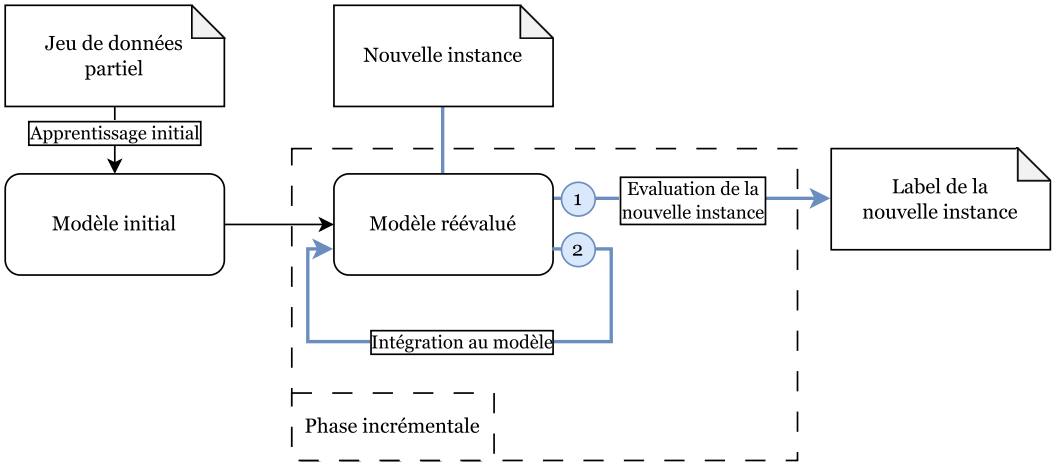

FIGURE 4.3 – Schéma représentant la simplification de la phase itérative de SuMeRI, décrite en Figure 4.1, à travers l'approche incrémentale. L'application continue est composée de deux étapes, répétées sur chaque nouvelle instance : (1) l'évaluation et (2) la mise à jour du modèle.

4.2.2 Définitions successives

Le second aspect de SuMeRI à adapter est l'application successive des méthodes. Naturellement, pour WOLF, il s'agit d'appliquer les méthodes associées aux différentes définitions successivement, ce qui correspond à l'objectif de ce chapitre.

A ce stade, deux possibilités sont considérées :

— réaliser une *application indépendante* des méthodes, équivalente à une application parallèle, sans que l'exécution d'une méthode n'ait d'impact sur celle des suivantes ;

— s'inspirer de l'approche successive de SuMeRI à travers une *approche jointe* où la sortie d'une méthode est utilisée comme entrée de la suivante, impactant ainsi les modèles successifs.

En référence à SuMeRI, l'adaptation est appelée *SuMeLI* pour Successive Methods Learned Incrementally.

4.2.2.1 Application indépendante : SuMeLInd

Dans le cas de l'application indépendante, le processus complet suit le schéma simple présenté en Figure 4.4. Il est référencé comme *SuMeLInd*.

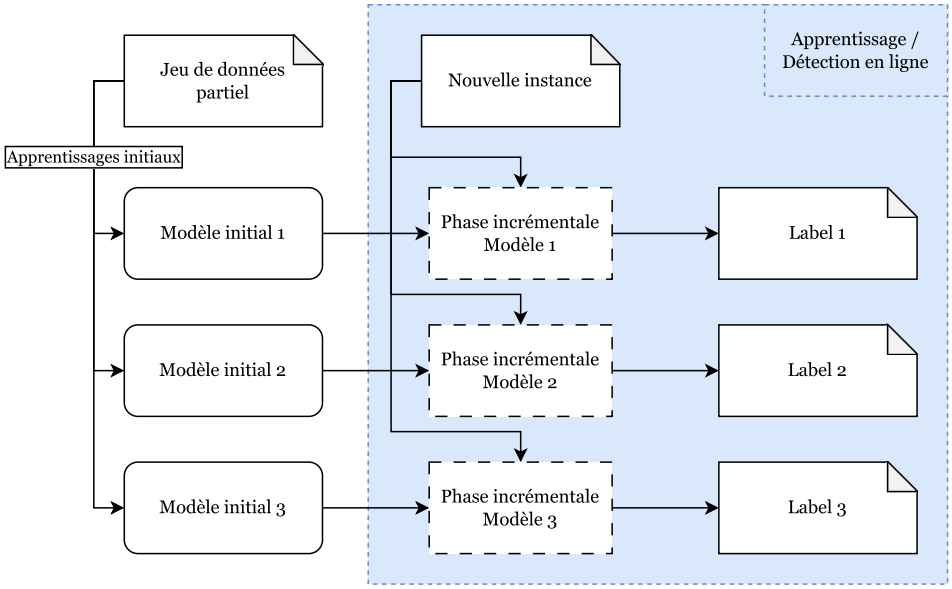

FIGURE 4.4 – Schéma représentant le processus de SuMeLInd.

4.2.2.2 Application jointe : SuMeLInk

Dans le cas de l'application jointe, référencée comme *SuMeLInk*, le processus est plus complexe puisque la sortie d'une méthode est utilisée comme entrée de la suivante.

L'apprentissage initial est le même que pour l'application indépendante, où les modèles sont appris en parallèle. Ainsi, les différentes méthodes peuvent partager le même modèle initial dans le cas de WOLF-KDE.

Dans SuMeRI, les modèles successifs sont appris à partir des données corrigées par les phases précédentes. De cette manière, les anomalies détectées par les phases précédentes ne sont pas rencontrées dans l'apprentissage. Pour SuMeLInk, une logique similaire est suivie : bien que toutes les instances soient évaluées par tous les modèles, la mise à jour des modèles n'est réalisée que pour les instances considérées comme normales par les phases précédentes.

Si cette approche a les mêmes avantages que l'application successive de SuMeRI, elle a cependant l'inconvénient de ralentir l'apprentissage de nouveaux comportements normaux par les derniers modèles de la liste. En effet, ces nouveaux comportements risquent d'être considérés pendant un certain temps comme anormaux par les premiers modèles avant d'être transmis aux modèles suivants. Néanmoins, cela signifie également que le temps de traitement peut être réduit en ne mettant pas tous les modèles à jour à chaque nouvelle instance observée.

La Figure 4.5 présente le processus complet de SuMeLInk. La première partie montre le processus incrémental avec prise en compte du pré-label établi dans les phases précédentes. Ce pré-label est (1) normal si l'instance est considérée comme normale dans toutes les phases précédentes et (2) anormal si l'instance est considérée comme anormale au moins une fois. La seconde partie reprend le schéma de SuMeLInd (Fig. 4.4) en montrant que la nouvelle instance traverse cette fois successivement les différents modèles.

4.2.3 Question du paramétrage

4.2.3.1 Paramétrage des méthodes

SuMeLI n'utilise plus l'apprentissage itératif. Cependant, l'objectif de cet apprentissage dans SuMeRI est de faciliter le paramétrage des méthodes en adaptant le modèle au fil des itérations, ce que ne permet donc pas SuMeLI. Aussi, il est important de choisir des méthodes avec un nombre de paramètres très faible ou nul. Idéalement, les paramètres doivent pouvoir être déduits et adaptés automatiquement à partir des données et de leur évolution.

En reprenant WOLF-KDE, proposé dans le chapitre précédent, les paramètres à fixer sont :

— pour la définition statistique : le seuil sur la pdf, la taille de la fenêtre (qui correspond au nombre de centres de noyaux), la fonction noyau et la méthode pour calculer la largeur de bande ;
— pour la définition de distance : le seuil k sur le nombre de voisins, le rayon de recherche R, la taille de la fenêtre, la fonction noyau et la méthode pour calculer la largeur de bande ;
— pour la définition de densité locale : le taux k_σ, la fraction $2^{-\alpha}$, le rayon de recherche R, la taille de la fenêtre, la fonction noyau et la méthode pour

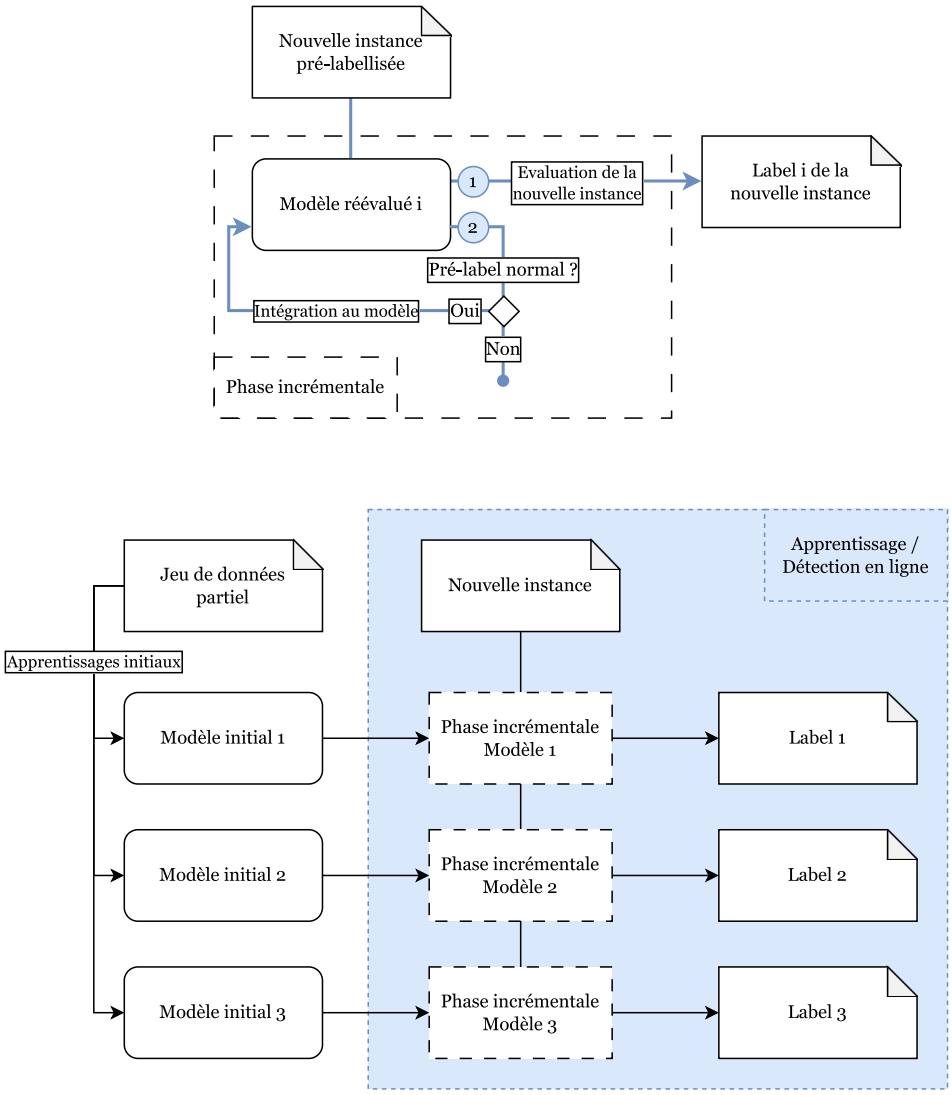

FIGURE 4.5 – Représentation du processus complet de SuMeLInk. Le premier schéma correspond à la phase incrémentale avec lien entre les méthodes et le second montre la différence avec la Figure 4.4 pour SuMeLInd.

calculer la largeur de bande.

Bien qu'il serait intéressant d'étudier d'autres possibilités, la fonction noyau et la méthode de calcul de la largeur de bande sont fixées dans cette étude. La taille de la fenêtre peut être fixée en fonction de l'espace mémoire disponible ou du temps de traitement désiré, et on prendra ici une fenêtre de 500 instances. Il est également recommandé de prendre $k_\sigma = 3$ et on prend $\alpha = 2$ comme dans les tests effectués dans le chapitre précédent.

Il reste ainsi trois paramètres à fixer : le seuil sur la pdf, le seuil sur le nombre de voisins et le rayon R. Dans les prochains tests, les deux premiers seront fixés manuellement. En revanche, R peut être fixé dynamiquement à partir des données en reprenant la méthode pour calculer la largeur de bande, évitant ainsi d'engendrer un sur-coût de calcul.

Il paraît cohérent que le rayon définissant le voisinage soit calculé comme un rapport au rayon de l'hyper-sphère circonscrite autour de l'hyper-cube supportant le noyau de Epanechnikov puisqu'il faut que l'hyper-cube soit entièrement compris dans le voisinage pour que le centre de noyau associé compte comme 1 voisin. Rappelons que pour chaque dimension i, le noyau de Epanechnikov est non-nul pour $\frac{\mathbf{x}_i}{\mathbf{H}_{i,i}} < 1$ avec $\mathbf{H}_{i,i} = \sqrt{5} n^{-\frac{1}{p+4}} \sigma_i$. Le rayon de l'hyper-sphère est ainsi $R' = \sqrt{\sum_{i=1}^{p} H_{i,i}^2}$. Ce rayon est cependant fortement influencé par les grandes valeurs de $H_{i,i}$ dans le cas où l'écart-type n'est pas équilibré sur toutes les dimensions. Pour R, il serait ainsi souhaitable d'utiliser une moyenne des $H_{i,i}$, et pour être proportionnelle au rayon de l'hyper-sphère, la moyenne quadratique, définie comme $\sqrt{\frac{\sum_{i=1}^{p} H_{i,i}^2}{n}}$, est la plus adaptée. Finalement, on définit ainsi $R = \beta \frac{R'}{\sqrt{p}}$ et on fixe $\beta = 0.5$ de manière arbitraire.

On sépare tout de même le rayon de la définition de distance $R_{dist} = R$ du rayon de la définition de densité locale R_{local}. En ce qui concerne la définition de densité locale, il semble plus adapté de prendre $r = R$, et donc $R_{local} = 2^\alpha R$, puisque les requêtes de voisinage sont en réalité majoritairement réalisées avec le rayon r.

4.2.3.2 Paramétrage de SuMeLI

Dans le cas de SuMeLInk, l'ordre d'application des méthodes est impactant puisque les anomalies détectées par les premières méthodes ne sont pas intégrées aux modèles des méthodes suivantes, son choix est donc important et doit être réfléchi.

Notons d'abord quelques liens logiques entre les différentes définitions dans le cadre de WOLF-KDE :

— une instance dont la valeur de la pdf associée est nulle est une anomalie statistique, elle peut cependant être normale selon les définitions de distance et de densité locale ; au contraire, une instance dont la valeur de la pdf associée n'est pas nulle a nécessairement un voisinage non-nul, et une instance n'ayant aucun voisin à R a une pdf associée nulle,

— une instance n'ayant aucun voisin dans son rayon R est à la fois anormale

selon la définition de distance et selon celle de densité locale ; en revanche, toutes les anomalies de distance ne sont pas des anomalies de densité locale puisque cela dépend des voisinages à R dans R_{local}, et c'est également vrai pour la réciproque.

Par ailleurs, la majorité des anomalies statistiques sont également des anomalies de distance, et généralement aussi de densité locale, et les tests du chapitre précédent ont démontré que la complexité augmente rapidement pour les méthodes basées distance et MDEF alors que la précision varie peu quand le nombre de centres de noyaux augmente. Ainsi, il semble plus judicieux de retenir l'ordre suivant dans SuMeLI, et qui sera utilisé dans la section suivante : définition statistique puis définition de distance et enfin définition de densité locale.

4.3 Comparaison des approches liée et indépendante

Cette section réalise la comparaison entre les performances de SuMeLInk et SuMeLInd, les deux approches présentées dans la section précédente, au sein de WOLF-KDE, défini par l'utilisation de trois méthodes de détection d'anomalies, basées sur l'estimation de densité par noyau (KDE), pour traiter respectivement trois définitions de l'anomalie (Définitions 5, 6 et 8). Cette comparaison est réalisée sur deux jeux de données synthétiques.

4.3.1 Présentation des jeux de données

4.3.1.1 Jeu des deux disques

Le premier jeu de données utilisé consiste en deux clusters sous forme de disques en deux dimensions. Les deux disques ont une densité différente : le premier contient 5000 éléments dans un cercle de faible rayon tandis que le second contient 1000 éléments dans un plus grand cercle. En plus de ces deux clusters, considérés comme normaux, 50 instances issues d'une distribution uniforme sont ajoutées pour représenter des anomalies.

Notons que, du fait de la répartition uniforme des anomalies, certaines peuvent se retrouver dans les clusters normaux, en particulier dans le cas du cluster épars. Pour éviter ce problème, un réajustement est effectué sur ces instances.

Le jeu des deux disques est représenté par la Figure 4.6, avec le cluster dense en bas à gauche et le cluster épars en haut à droite.

4.3.1.2 Jeu des deux lunes

Le second jeu de données consiste également en deux clusters en deux dimensions, mais prenant cette fois-ci la forme de deux lunes opposées. La densité des deux clusters est cette fois la même, mais l'entrelacement des lunes rend la modélisation de la normalité plus complexe. Chaque lune contient 2500 échantillons et 20 instances anormales sont ajoutées, modélisées par une distribution uniforme.

Le jeu des deux lunes est représenté par la Figure 4.7.

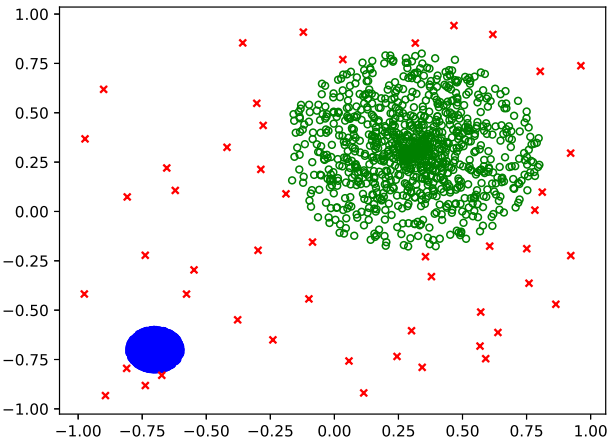

FIGURE 4.6 – Jeu de données des deux disques avec le cluster dense en bleu, le cluster épars en vert et les anomalies en rouge.

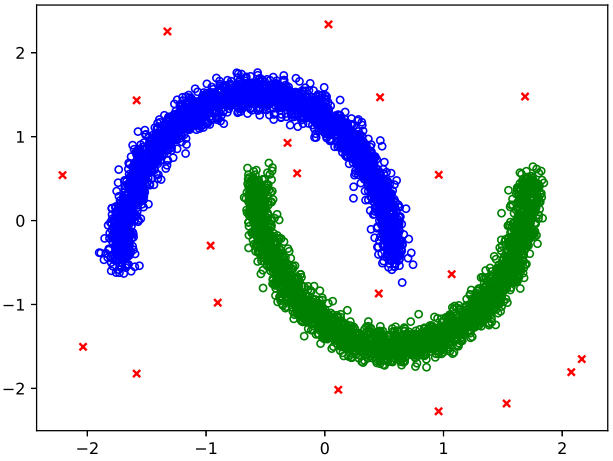

FIGURE 4.7 – Jeu de données des deux lunes avec la première lune en bleu, la seconde en vert et les anomalies en rouge.

4.3.1.3 Ordre des clusters

Les deux jeux de données mentionnés sont utilisés selon deux variantes :

— selon la première variante, appelée *variante aléatoire*, toutes les instances sont mélangées de manière aléatoire dans le jeu de données, ce qui revient à dire que la distribution des données n'évolue pas dans le temps et correspond au mélangé de trois distributions différentes (une pour chaque cluster normal et une distribution uniforme) avec des poids différents,

— selon la seconde variante, appelée *variante successive*, une première partie du jeu de données ne contient que des instances du premier cluster tandis qu'une seconde ne contient que des instances du second cluster, et les anomalies sont réparties aléatoirement au sein de l'ensemble du jeu de données ; cette

variante correspond à un changement net de la distribution au cours de la passe sur les données.

4.3.2 Présentation des résultats

Cette sous-section présente les résultats obtenus sur les jeux de données et leurs variantes. Dans un premier temps, les métriques d'évaluation utilisées sont expliquées, puis les paramètres choisis sont présentés avant de finalement restituer les résultats.

4.3.2.1 Métriques utilisées

Les jeux de données utilisés étant synthétiques, un label est disponible pour cette évaluation qui peut donc être supervisée. Aussi, la méthode d'évaluation non supervisée WOLF-Eval n'est pas utilisée.

Différentes métriques usuelles de l'évaluation supervisée ont été présentées dans la Section 1.4.1.1. Ici sont retournés : le *rappel*, mesurant le taux d'anomalies correctement classées, la *précision*, mesurant le taux d'anomalies réelles parmi les instances classées anormales et la *F-mesure*, donnant la moyenne harmonique des deux métriques précédentes.

En plus de ces trois métriques, qui nécessitent de fixer un seuil sur le score d'anomalie, deux autres métriques sont calculées : l'AUROC et l'AP (qui estime l'AUPRC). Ces métriques ont été présentées en Section 1.4.2.2 comme couramment utilisées car elles sont calculées sur une plage de seuils. L'AUROC est considérée comme trop optimiste par rapport à l'AUPRC mais a l'avantage d'être interprétable comme la probabilité qu'une anomalie ait un score plus élevé qu'une instance normale.

Les trois premières métriques sont calculées (1) séparément pour les labels des trois méthodes et (2) de manière globale en considérant une anomalie globale comme une instance anormale pour *au moins deux méthodes*. Ce second choix permet d'étudier une possible combinaison des décisions en une seule, ce qui simplifierait la lecture de la sortie par un opérateur. Cependant, puisqu'on perd ainsi l'interprétation des labels assistée par les définitions d'anomalies, les labels séparés conservent un intérêt majeur.

Pour les deux dernières métriques, on ne dispose pas d'un score global pouvant être utilisé pour décider des seuils, donc elles ne sont données que pour les labels des trois méthodes et pas pour le label global.

Quatre graphiques sont finalement fournis : un pour les anomalies statistiques, un pour les anomalies de distance, un pour les anomalies de densité locale et un pour les anomalies globales comme définies précédemment.

4.3.2.2 Choix des paramètres

Comme mentionné dans la section précédente, il reste deux paramètres à fixer qui correspondent aux seuils pour la méthode statistique et la méthode basée dis-

tance.

Une approche commune pour fixer le seuil en présence de labels est l'utilisation des courbes ROC et rappel-précision. La première trace le rappel (R) en fonction du taux de faux positif (FPR) ; un seuil maximisant le rappel maximise le taux de faux positif tandis qu'un seuil minimisant le taux de faux positif minimise le rappel. Une méthode parfaite, avec le meilleur paramètre de seuil, permet d'obtenir un rappel de 1 pour un taux de faux positif de 0, et l'intérêt de l'utilisation de la courbe ROC est de trouver la valeur de seuil correspondant au point de la courbe le plus proche de $(0, 1)$. La seconde courbe trace la précision (P) en fonction du rappel et son utilisation suit la même logique sauf que la courbe associée à une méthode parfaite atteint le point $(1, 1)$.

Trois seuils sont sélectionnés indépendamment pour chacune des deux méthodes et sur les deux jeux de données et leurs deux dispositions différentes :

— le seuil maximisant la valeur $R \times (1 - FPR)$;
— le seuil maximisant la valeur $P \times R$;
— le seuil maximisant le produit des deux valeurs précédentes.

Pour chaque jeu de données, les tests ont été effectués pour les 9 combinaisons de seuils ainsi obtenus, et on choisit de ne retourner que la combinaison donnant les meilleurs résultats de F-mesure moyenne (i.e. : la moyenne de F-mesure pour SuMeLInk et SuMeLInd pour les labels en sortie des trois méthodes et pour les anomalies globales). Les seuils ainsi obtenus sont restitués dans le Tableau 4.1 :

Jeu de données	Statistique	Distance [1]
Deux disques (variante aléatoire)	$t = 0.04$	$k = 2$
Deux disques (variante successive)	$t = 0.03$	$k = 0.01$
Deux lunes (variante aléatoire)	$t = 0.03$	$k = 3$
Deux lunes (variante successive)	$t = 0.02$	$k = 0.3$

TABLE 4.1 – Paramètres retenus pour les méthodes statistique et basée distance et pour les différents jeux de données

Il faut tout de même noter que ces paramètres n'impactent que les trois premières métriques mais pas l'AUROC et l'AP.

4.3.2.3 Résultats

Les résultats obtenus sont restitués dans les Figures 4.8, 4.9, 4.10 et 4.11. Les 500 premières instances de chaque jeu de données sont utilisées pour l'entraînement ; ces résultats sont ainsi calculés sur l'évaluation en ligne des instances suivantes.

Chaque figure est composée de quatre graphiques. Le premier, en haut à gauche, donne les résultats à partir des labels obtenus grâce à la méthode statistique, le second, à sa droite, grâce à la méthode basée distance, le troisième, en bas à gauche, grâce à la méthode basée sur le MDEF et le dernier à partir du label global proposé.

Pour le premier graphique, la comparaison entre les approches SuMeLInk et SuMeLInd n'apporte pas d'information puisque, à ce stade, les résultats sont les

mêmes. Aussi, comme discuté précédemment, les métriques AUROC et AP ne sont pas fournies pour le graphique global.

Analyse pour le jeu des deux disques Les performances obtenues pour les deux approches sont globalement satisfaisantes, à l'exception de l'évaluation de la méthode basée sur le MDEF dans le cas de la variante aléatoire. Les deux approches permettent tout de même d'obtenir de bons résultats sur un jeu de données multi-densité.

Pour la variante aléatoire (Fig. 4.8), les performances pour les différentes métriques sont légèrement meilleures avec SuMeLInk, démontrant l'intérêt de la stratégie proposée. Les faibles performances de la méthode basée sur le MDEF sont justifiées par le faible nombre d'instances considérées comme anormales ; toutes les anomalies prédites sont bien des anomalies, mais le pourcentage d'anomalies détectées reste faible.

Pour la méthode globale, on remarque que le rappel est le même que pour la méthode basée distance avec une plus grande précision ; cela s'explique par le fait que les anomalies détectées par la méthode basée distance sont aussi détectées par au moins l'une des deux autres méthodes. Ainsi, ces anomalies sont considérées comme des anomalies globalement avec l'approche proposée, et avec la double validation qui découle de la définition des anomalies globales, le nombre de faux positif est également réduit, augmentant la précision.

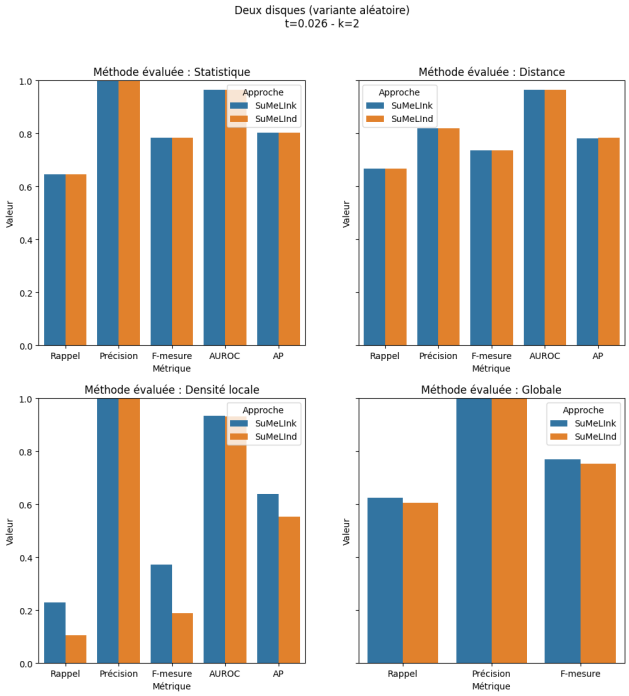

FIGURE 4.8 – Résultats obtenus pour SuMeLInk et SuMeLInd sur la variante aléatoire du jeu des deux disques

Pour la variante successive (Fig. 4.9), les performances sont cette fois meilleures pour l'approche SuMeLInd, sauf dans le cas du rappel. Cette observation valide la limite identifiée concernant le fait que SuMeLInk induit un délai dans l'apprentissage d'un nouveau comportement normal. En effet, avec la variante successive, les clusters normaux arrivent successivement et les modèles doivent apprendre le nouveau comportement.

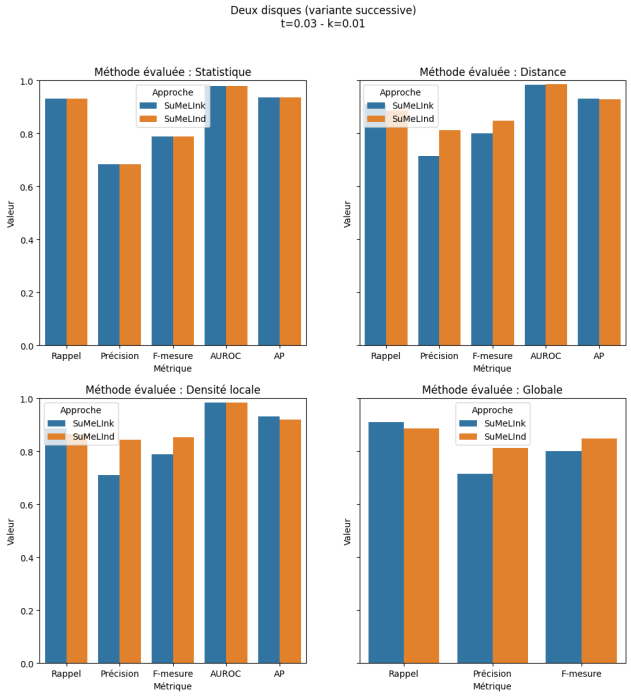

FIGURE 4.9 – Résultats obtenus pour SuMeLInk et SuMeLInd sur la variante successive du jeu des deux disques

L'étude des métriques AUROC et AP démontre également que les performances sont meilleures pour la variante successive mais aussi que, pour ces métriques, SuMeLInk a des performances similaires à SuMeLInd, parfois même légèrement meilleures. Ceci s'explique par le fait que les paramètres optimaux ont été choisis à partir de l'application indépendante des méthodes, et ces paramètres n'impactent que le rappel et la précision, pas l'AUROC et l'AP. Cette observation témoigne de l'importance du choix des paramètres.

Analyse pour le jeu des deux lunes Les performances pour le jeu des deux lunes sont moins bonnes que dans le cas des deux disques. Bien que les clusters soient aussi denses, leur forme est plus complexe et impacte la précision des modèles. On remarque également, pour la densité locale et avec la variante aléatoire, le cas où aucune anomalie n'est détectée ; il n'y a donc ni vrais positifs, ni faux positifs dans ce cas, et le rappel est nul et la précision non définie.

Pour la variante aléatoire (Fig. 4.10), les performances sont les mêmes pour les

deux approches selon les trois premières métriques. Le rappel étant le même pour les anomalies de distance et globales, on peut conclure que toutes les anomalies de distance sont ici des anomalies statistiques. Aussi, la précision étant égale à 1, toutes les anomalies de distance sont de vraies anomalies, mais environ 60% d'entre elles ne sont pas détectées.

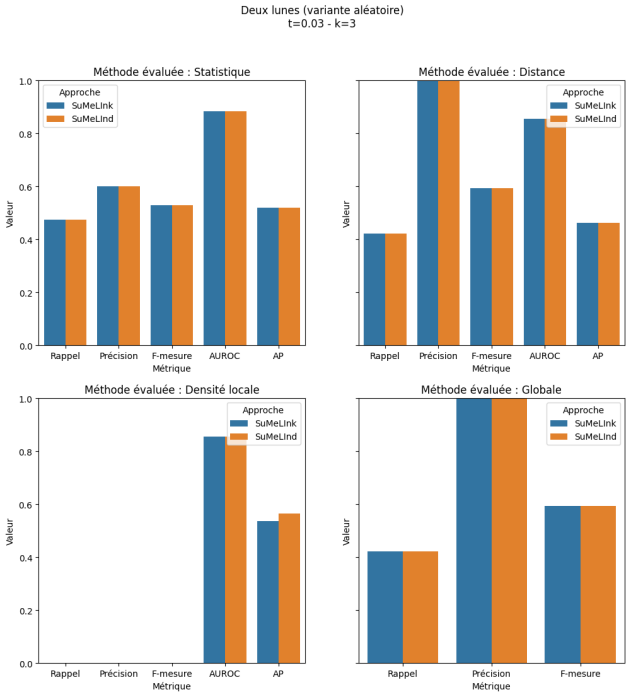

FIGURE 4.10 – Résultats obtenus pour SuMeLInk et SuMeLInd sur la variante aléatoire du jeu des deux lunes

Les résultats sont plus mauvais pour la variante successive (Fig. 4.11) à cause du changement de cluster qui provoque la détection de faux positifs, réduisant la précision et en particulier pour SuMeLInk à cause de la latence induite par l'approche dans l'apprentissage d'un nouveau comportement normal.

Bien que les performances selon l'AUROC soient similaires à ce que l'on pouvait observer sur le jeu de données des deux disques, les performances selon l'AP sont elles assez différentes. Premièrement, les performances en AP sont moins bonnes sur ce jeu de données de manière générale. Mais on remarque surtout une baisse de performance pour SuMeLInk et pour la variante successive. Cette baisse de performance, qui devient plus importante avec la succession des méthodes, témoigne bien de l'effet attendu concernant le délai d'apprentissage des nouveaux comportements.

Enfin, les performances de l'AUROC et l'AP pour la densité locale, principalement dans le cas de la variante aléatoire, mettent en évidence la faiblesse du seuil fixé automatiquement. En effet, la valeur de ces métriques semble suffisamment élevée pour conclure qu'un ajustement manuel aurait pu grandement améliorer les résultats en rappel et précision. Cependant, l'approche globale corrige en partie ce

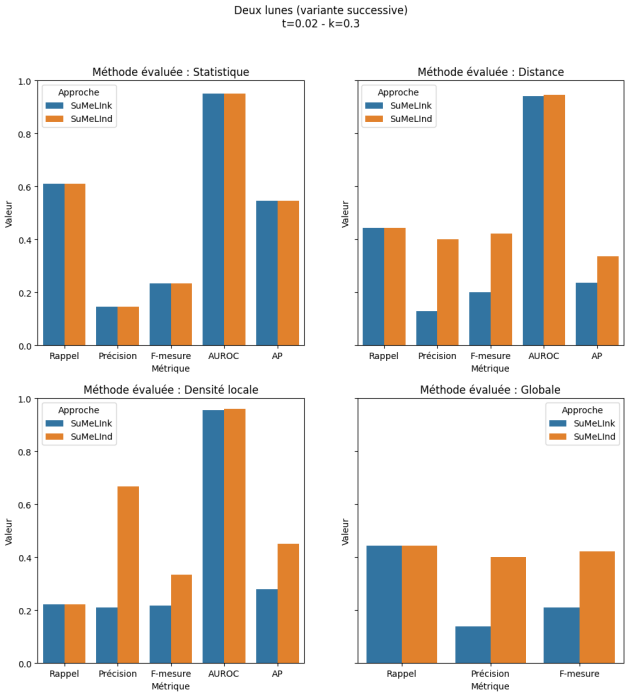

FIGURE 4.11 – Résultats obtenus pour SuMeLInk et SuMeLInd sur la variante successive du jeu des deux lunes

trait puisqu'il suffit d'observer une anomalie selon deux définitions pour considérée qu'une instance est globalement anormale, réduisant l'impact de la méthode de densité locale. Aussi, les valeurs faibles d'AP pour la variante successive témoigne également de la faiblesse des méthodes pour ce jeu de données.

Analyse générale Si SuMeLInk peut parfois offrir de meilleures performances que SuMeLInd, de manière générale l'approche induira une perte de performances en cas de changement dans la distribution.

Ces expérimentations ont aussi permis de mettre en évidence la sensibilité du paramétrage du seuil. Premièrement, pour les mêmes jeux de données, modifier l'ordre des instances modifie les paramètres optimaux obtenus. Ensuite, le seuil automatique fourni par le MDEF semble peu adapté et, comme en témoignent les résultats plus probants selon l'AUROC et l'AP, il faudrait le fixer manuellement, ce qui n'est cependant pas souhaitable dans un cadre non supervisé.

4.4 Bilan de l'étude

Cette dernière section réalise le bilan des différents éléments présentés dans ce quatrième chapitre.

4.4.1 Adaptation de SuMeRI

Dans ce chapitre est proposée une adaptation de SuMeRI, présenté dans [Ducharlet 2020], pour une application en ligne dans WOLF-KDE, le cadre unifié du Chapitre 3E. Cette adaptation, appelée SuMeLI pour Successive Methods Learned Incrementally, a été déclinée sous deux approches :

— SuMeLInk, qui relie les méthodes appliquées successivement en restreignant la mise à jour des modèles aux instances considérées normales dans les phases précédentes,

— SuMeLInd, qui exécute les méthodes indépendamment les unes des autres.

Alors que SuMeRI offrait la possibilité de combiner des méthodes pour détecter plusieurs types d'anomalies dans un contexte hors ligne, les deux déclinaisons de SuMeLI permettent de combiner les définitions du cadre opérationnel WOLF pour une application en ligne.

4.4.2 Limites de SuMeLI

L'étude des performances de SuMeLInk et SuMeLInd sur WOLF-KDE et leur comparaison a mis en évidence les limites de SuMeLI. La mise en relation des méthodes dans SuMeLInk réduit ses performances par rapport à SuMeLInd dans le cas d'un changement de comportement en réduisant la faculté des méthodes à apprendre ce nouveau comportement. On peut également noter qu'avec une application indépendante comme proposée par SuMeLInd, l'exécution successive peut être remplacée par une exécution parallèle si les capacités du capteur le permettent, réduisant le temps de traitement. D'un autre côté, SuMeLInk doit être légèrement plus performant en exécution successive puisque toutes les instances ne sont pas intégrées, ce qui peut réduire l'occupation mémoire des modèles. Cependant, ce n'est pas le cas pour WOLF-KDE ; en effet, puisque le modèle est partagé, l'occupation mémoire est réservée au plus grand, et il faut même ajouter une liste conservant en mémoire les indices des centres de noyaux utilisés pour les modèles de distance et de densité locale.

De plus, bien que les performances de SuMeLI soient correctes pour les métriques AUROC et AP, elles le sont moins pour les métriques reposant sur un seuil fixé sur le score. Or, dans un cadre d'une application entièrement non supervisée, les paramètres seuil doivent être fixés manuellement ou automatiquement. La sensibilité des méthodes à ces paramètres est donc une limite majeure de SuMeLI, et plus généralement de WOLF-KDE.

4.4.3 Discussions

Plusieurs points peuvent être discutés à la lumière des éléments mis en avant dans ce chapitre :

— combiner les différentes définitions de WOLF n'est pas évident ; l'approche évidente de SuMeLInd, qui prend les résultats des différentes méthodes indépendamment apparait comme la plus sûre. De plus, dans un cadre plus

général, choisir comment appliquer les méthodes successivement n'est pas évident, et les limites de SuMeLInk devraient s'amplifier en multipliant le nombre de définitions ;

— le problème du paramétrage reste une limite critique ; les méthodes à base de KDE sont sensibles au paramétrage du seuil sur le score et il nous semble par conséquent important de chercher des méthodes plus robustes pour les remplacer.

Chapitre 5

La fonction de Christoffel pour la détection d'anomalies

Dans les précédents chapitres, la méthode d'estimation de densité par noyau (KDE) a été utilisée pour la détection d'anomalies au sein du cadre opérationnel WOLF sous le nom WOLF-KDE. Si les résultats obtenus étaient concluants, ils démontraient tout de même la difficulté à paramétrer les méthodes.
Ce cinquième chapitre propose d'utiliser une nouvelle méthode dont le paramétrage est simplifié et satisfaisant les contraintes des réseaux de capteurs. Cette méthode, appelée *DyCF* pour *Dynamical Christoffel Function*, repose sur le calcul de la *Fonction de Christoffel Empirique (FCE)* et fait l'objet d'une soumission sous le format d'article de revue. Elle est intégrée dans le cadre opérationnel WOLF, sous la forme *WOLF-DyCF*, et comparée à WOLF-KDE. La comparaison est réalisée sur les données du Chapitre 3 et sur un jeu de données industriel présenté en Section 1.5.2.

Sommaire

5.1 Proposition de méthodes basées sur la Fonction de Christoffel Empirique

La première section de ce chapitre présente la Fonction de Christoffel Empirique (FCE), ses origines et quelques unes de ses propriétés, utilisées pour définir une nouvelle méthode de détection d'anomalies applicables aux flux de données, et plus spécifiquement aux WSNs : *DyCF* (pour *Dynamical Christoffel Function*).

5.1.1 Introduction au noyau de Christoffel-Darboux

Le *Noyau de Christoffel-Darboux* (NCD), et la *Fonction de Christoffel* (FC) associée, proviennent de la théorie de l'approximation et des polynômes orthogonaux [Nevai 1986, Dunkl 2001]. Ils ont cependant été longtemps ignorés en analyse de données discrètes alors que des résultats récents ont démontré que certaines de leurs propriétés pourraient avoir un intérêt majeur en science des données, notamment en détection d'anomalies [Lasserre 2019, Lasserre 2022].

Le NCD et la FC sont associés à une mesure μ, qu'on supposera ici être une mesure de Borel et une mesure de probabilité, supportée par $\Omega \subset \mathbb{R}^p$ que l'on considère compact et d'intérieur non vide. Ils sont également paramétrés par d et intimement liés à la matrice des moments de la mesure μ, indexée par une base orthonormée de l'ensemble $\mathbb{R}_d[\mathbf{x}]$, avec $\mathbf{x} = (x_1, x_2, ..., x_p) \in \mathbb{R}^p$, des polynômes de degrés au plus d, de dimension $s_p(d) = \binom{p+d}{d}$.

On adopte la notation multi-index pour les polynômes, et en posant le vecteur des degrés $\alpha = (\alpha_1, \alpha_2, ..., \alpha_p) \in \mathbb{N}^p$, on définit $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} ... x_p^{\alpha_p}$, avec un degré total noté $deg(\alpha) = \sum_{i=1}^{p} \alpha_i$.

Soit $\mathbf{v}_d : \mathbb{R}^p \mapsto \mathbb{R}^{s_p(d)}$ un vecteur dont les éléments correspondent à la base orthonormée retenue pour $\mathbb{R}_d[\mathbf{x}]$, la matrice des moments de μ associée est une matrice symétrique réelle, semi-définie positive, définie par :

$$M_{\mu,d} = \int_{\mathbb{R}^p} \mathbf{v}_d(\mathbf{x}) \mathbf{v}_d(\mathbf{x})^T d\mu(\mathbf{x}) \in \mathbb{R}^{s_p(d) \times s_p(d)}, \tag{5.1}$$

où l'intégrale est calculée composant par composant.

On peut par exemple choisir la base des monômes, et dans ce cas $\mathbf{v}_d(\mathbf{x}) = (\mathbf{x}^{\alpha_i})_{1 \leq i \leq s_p(n)}$ où les α_i correspondent aux combinaisons de degrés vérifiant $deg(\alpha_i) \leq d$, ordonnés :

- d'abord par degré total $deg(\alpha_i)$ croissant,
- puis, pour un même degré total, selon les $\alpha_{i,j}$ successifs, avec $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,p})$; c'est-à-dire que α_i précède α_j si $\alpha_{i,1} > \alpha_{j,1}$, et si $\alpha_{i,1} = \alpha_{j,1}$ alors α_i précède α_j si $\alpha_{i,2} > \alpha_{j,2}$, et ainsi de suite, ce qui donne un ordre lexicographique sur les monômes.

Aussi, pour $d = 2$ et $p = 2$, on a :

$$\mathbf{v}_d(\mathbf{x}) = (\mathbf{x}^{(0,0)}, \mathbf{x}^{(1,0)}, \mathbf{x}^{(0,1)}, \mathbf{x}^{(2,0)}, \mathbf{x}^{(1,1)}, \mathbf{x}^{(0,2)}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2),$$

et la matrice des moments est donnée par :

$$M_{\mu,2}: \begin{array}{c|cccccc} & 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ \hline 1 & 1 & y_{1,0} & y_{0,1} & y_{2,0} & y_{1,1} & y_{0,2} \\ x_1 & y_{1,0} & y_{2,0} & y_{1,1} & y_{3,0} & y_{2,1} & y_{1,2} \\ x_2 & y_{0,1} & y_{1,1} & y_{0,2} & y_{2,1} & y_{1,2} & y_{0,3} \\ x_1^2 & y_{2,0} & y_{3,0} & y_{2,1} & y_{4,0} & y_{3,1} & y_{2,2} \\ x_1x_2 & y_{1,1} & y_{2,1} & y_{1,2} & y_{3,1} & y_{2,2} & y_{1,3} \\ x_2^2 & y_{0,2} & y_{1,2} & y_{0,3} & y_{2,2} & y_{1,3} & y_{0,4} \end{array} \tag{5.2}$$

avec $y_{i,j} = \int_{\mathbb{R}^p} \mathbf{x}^{(i,j)} d\mu(\mathbf{x})$.

On définit ensuite le NCD à partir de la matrice des moments :

$$K_d^\mu : (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{v}_d(\mathbf{x})^T M_{\mu,d}^{-1} \mathbf{v}_d(\mathbf{y}) \,, \tag{5.3}$$

et la FC en découle par la relation :

$$\Lambda_d^\mu(\mathbf{x}) = \frac{1}{K_d^\mu(\mathbf{x}, \mathbf{x})} \,. \tag{5.4}$$

Notons que $Q_d^\mu(\mathbf{x}) = K_d^\mu(\mathbf{x}, \mathbf{x}) = 1/\Lambda_d^\mu(\mathbf{x})$ est un polynôme de degré $2d$ défini comme la somme de carrés de polynômes.

L'une des propriétés les plus importantes de la FC est sa capacité à décrire le support Ω de μ. En particulier, la courbe de niveau décrite par l'ensemble $S_\gamma := \{\mathbf{x} : \Lambda_d^\mu(\mathbf{x}) \geq \gamma\}$, pour un $\gamma \in \mathbb{R}_+$ bien choisi, capture de manière assez précise la forme de Ω, même pour des valeurs faibles du paramètre d.

5.1.2 Calcul de la fonction de Christoffel empirique

Le NCD et la FC ont été mis en avant par J.-B Lasserre et E. Pauwels [Lasserre 2019, Lasserre 2022] comme un puissant outil pour l'analyse de données permettant d'approcher la densité, d'inférer le support ou de détecter des anomalies à partir d'un échantillonnage fini de la mesure μ. Dans ce cas, la matrice des moments ne peut être calculée que de manière empirique à partir d'un échantillon de N instances $\mathcal{X} = \{\mathbf{x}_i, 0 \leq i \leq N\}$ qui constitue la mesure discrète associée μ_N et de support $\mathcal{X}$, définie comme :

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x}_i} \tag{5.5}$$

où $\delta_{\mathbf{x}_i}$ correspond à la mesure de Dirac de support $\mathbf{x}_i$.

En reprenant l'Équation 5.1, on peut alors définir la matrice des moments empirique :

$$M_{\mu_N,d} = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbf{v}_d(\mathbf{x}_i) \mathbf{v}_d(\mathbf{x}_i)^T \,, \tag{5.6}$$

et la Fonction de Christoffel Empirique (FCE) :

$$\Lambda_d^{\mu_N}(\mathbf{x}) = \frac{1}{\mathbf{v}_d(\mathbf{x})^T M_{\mu_N,d}^{-1} \mathbf{v}_d(\mathbf{x})}\,. \tag{5.7}$$

En passant de la mesure μ de support Ω à μ_N dont le support est un échantillon $\mathcal{X}$, il est nécessaire de choisir d convenablement en fonction de N pour que $\Lambda_d^{\mu_N}$ partage les mêmes propriétés que la fonction de population Λ_d^{μ}. En effet, la similarité entre les deux fonctions provient essentiellement de la loi des grands nombres, comme décrit dans [Lasserre 2022] (§6.2), et il suffit donc que N soit suffisamment grand par rapport à d.

Il est toutefois important de noter qu'avoir N suffisamment grand n'implique pas un coût calculatoire plus important puisque la complexité du calcul de la FCE ne dépend que de la taille de la matrice des moments qui ne dépend elle-même que de p et d. $M_d(\mu_N)$ peut alors être perçue comme un encodage du jeu de données $\mathcal{X}$, ce qui permet à la FCE de respecter les contraintes en mémoire des modèles dans les WSNs sans utiliser de fenêtres glissantes. De plus, la définition de la matrice des moments empiriques implique la possibilité de combiner plusieurs matrices, avec $N_g M_{\mu_{N_g},d} = N_1 M_{\mu_{N_1},d} + N_2 M_{\mu_{N_2},d}$, respectant donc également la contrainte sur la combinaison des modèles.

Le paramètre d joue un rôle de compromis entre la régularité des courbes de niveaux décrites par les S_γ et leur ajustement au support de la mesure empirique ; pour d suffisamment élevé, celles-ci décrivent précisément les éléments de $\mathcal{X}$. Aussi, en choisissant convenablement la valeur γ, on peut considérer que toutes les instances à l'extérieur de l'ensemble de niveau donné par les $\mathbf{x}$ vérifiant $\Lambda_d^{\mu_N}(\mathbf{x}) < \gamma$ sont des anomalies tandis que les instances à l'intérieur sont normales.

En particulier, [Vu 2020] montre que, sous certaines conditions sur la mesure et en choisissant convenablement d en fonction de $r \geq 0$, $0 < \varepsilon < 1$ et N, on a que le seuil défini par :

$$\gamma_{\varepsilon,r} = 12 \left(\frac{3p(2-\varepsilon) + 3(1-\varepsilon)r}{2\varepsilon e} \right)^{\frac{p(2-\varepsilon)+(1-\varepsilon)r}{\varepsilon}} \frac{1}{d^{p(2-\varepsilon)+(1-\varepsilon)r}} \tag{5.8}$$

permet à l'ensemble $S_{\gamma_{\varepsilon,r}}$ associé de converger vers le support de la mesure quand N tend vers l'infini, avec ε assurant un meilleur taux de convergence quand il est pris petit et r une constante liée à la mesure μ.

En suivant [Lasserre 2022] (Théorème 7.3.3) et en posant $\varepsilon = 1/2$ et $r = 0$ dans l'Équation 5.8, on fixe :

$$\gamma_C := C d^{-3p/2}\,. \tag{5.9}$$

Aussi, en posant

$$\mathbf{S}_{d,\gamma_C}(\mathbf{x}) := \gamma_C / \Lambda_d^{\mu_N}(\mathbf{x}) = \gamma_C Q_d^{\mu_N}(\mathbf{x})\,, \tag{5.10}$$

on obtient que $\mathbf{S}_{d,\gamma_C}$ est une fonction de score d'anomalies bien définie, avec un seuil à 1 pour séparer les instances normales des instances anormales. On a ainsi défini

une méthode de détection d'anomalies basée sur l'inverse de la FCE, avec comme paramètres le degré d et la constante C. C influence le seuil et est dépendant du problème. Dans cette étude, on choisit de fixer $C = 1$ et on utilise donc la fonction de score $\mathbf{S}_{d,\gamma_1}(\mathbf{x})$.

5.1.3 DyCF : forme incrémentale de la FCE

Pour pouvoir appliquer la méthode proposée aux WSNs, il est nécessaire de pouvoir calculer la matrice des moments de manière incrémentale afin de ne pas avoir à conserver en mémoire l'ensemble des instances. Or, d'après l'Équation 5.6, on a :

$$M_{\mu_{N+1},d} = NM_{\mu_N,d} + \mathbf{v}_d(\mathbf{x}_{N+1})\mathbf{v}_d(\mathbf{x}_{N+1})^T\,, \tag{5.11}$$

ce qui amène naturellement au calcul incrémental. Il est ainsi possible de mettre à jour le modèle avec de nouvelles instances tout en maintenant un coût relativement faible.

Notons cependant que le calcul du score $\mathbf{S}_{d,\gamma_1}(\mathbf{x})$ d'une instance nécessite l'inversion de la matrice des moments. On peut toutefois utiliser la formule de Sherman-Morrison :

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}\,, \tag{5.12}$$

pour incrémenter la matrice inverse directement, évitant ainsi de réaliser l'inversion à chaque nouvelle instance. Ceci donne :

$$M^{-1}_{\mu_{N+1},d} = (N+1)\Big[(NM_{\mu_N,d})^{-1} - \frac{(NM_{\mu_N,d})^{-1}\mathbf{v}_d(\mathbf{x}_{N+1})\mathbf{v}_d(\mathbf{x}_{N+1})^T(NM_{\mu_N,d})^{-1}}{1 + \mathbf{v}_d(\mathbf{x}_{N+1})^T(NM_{\mu_N,d})^{-1}\mathbf{v}_d(\mathbf{x}_{N+1})}\Big]\,. \tag{5.13}$$

La méthode de détection d'anomalie basée sur le calcul du score $\mathbf{S}_{d,\gamma_1}(\mathbf{x})$ et l'incrémentation de la matrice des moments à chaque nouvelle instance est référencée par la suite comme *DyCF* pour *Dynamical Christoffel Function*.

5.2 Intégration dans WOLF

L'objectif de cette seconde section est d'étudier l'implémentation de DyCF au sein du cadre opérationnel WOLF proposé dans le Chapitre 3.

5.2.1 Rapport à la densité de probabilité

Par son lien avec la mesure de probabilité μ associée à la distribution des données, DyCF peut être considérée comme une méthode statistique.

Si elle n'approxime pas directement la pdf, comme c'est le cas d'une KDE, la FCE a tout de même un comportement similaire. Pour illustrer ce point, prenons trois distributions statistiques usuelles en une dimension : une distribution uniforme

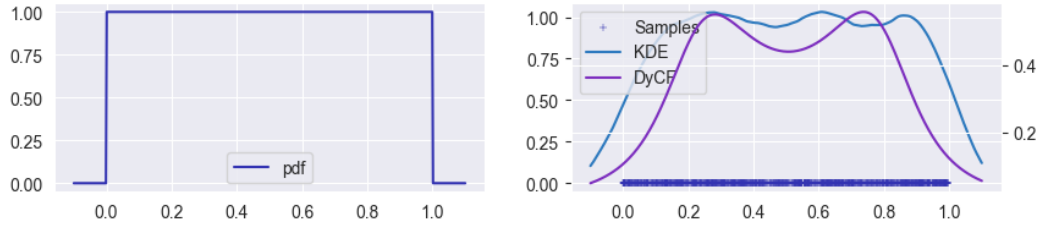

FIGURE 5.1 – Illustration de l'hypothèse d'approximation de la pdf par DyCF avec une distribution uniforme

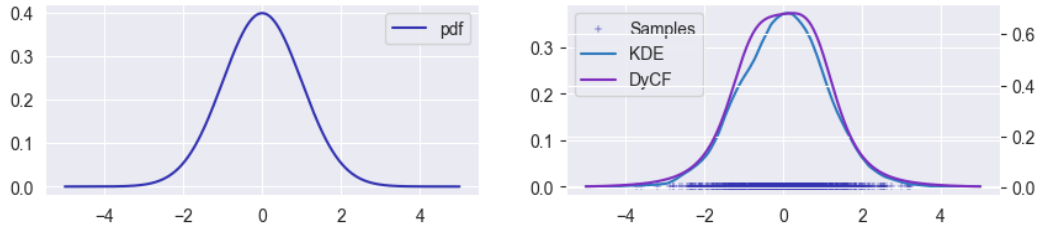

FIGURE 5.2 – Illustration de l'hypothèse d'approximation de la pdf par DyCF avec une distribution normale

de support $[-1, 1]$, une distribution normale avec $\mu = 0$ et $\sigma^2 = 1$ et une distribution bêta avec $\alpha = 2$ et $\beta = 5$. Pour ces trois distributions, on génère un ensemble de 1000 échantillons et on compare l'estimation de la pdf par KDE et la FCE, en entraînant les modèles avec ces échantillons, à la pdf réelle associée à la distribution. Les résultats pour ces trois distributions sont présentés respectivement sur les Figures 5.1, 5.2 et 5.3. On y remarque une similitude de l'approximation dans son comportement, en particulier pour la distribution normale et la distribution bêta.

Ainsi, il semble intéressant d'évaluer la détection d'anomalies de DyCF avec WOLF-Eval et selon la définition statistique fournie dans WOLF (Définition 5).

Rappelons que WOLF-Eval nécessite de disposer du score d'anomalie théorique et du classement des instances à la fois selon ce score théorique et selon celui obtenu par les méthodes évaluées. Pour la définition statistique, le score d'anomalie utilisé

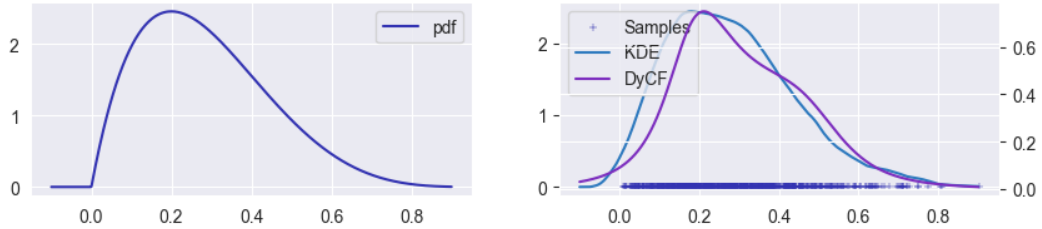

FIGURE 5.3 – Illustration de l'hypothèse d'approximation de la pdf par DyCF avec une distribution bêta

pour les classements repose sur la pdf f, ou son approximation $\tilde{f}$, et on prend précisément $1/(1+f)$ comme score d'anomalie.

Dans le cas de DyCF, le comportement de f est estimé par celui de $\Lambda_d^{\mu_N}$, à valeur dans $\mathbb{R}_+^*$, et on peut donc utiliser $\mathbf{S}_{d,\gamma_1}$ pour obtenir un classement similaire à $1/(1+f)$. De ce fait, DyCF peut directement être comparée à la méthode par KDE sur la définition statistique.

Puisque le score renvoyé par DyCF suit un comportement similaire à celui renvoyé par la méthode par KDE, il parait également intéressant d'estimer, de manière similaire, les anomalies de distance et de densité locale. On peut ainsi définir un cadre unifié *WOLF-DyCF*, directement comparable à WOLF-KDE car appliqué aux mêmes définitions de l'anomalie.

5.2.2 Définition statistique

Pour réaliser l'évaluation de DyCF sur la définition statistique, la démarche suivie reprend celle de la comparaison de la précision des méthodes par KDE et GMM de la Section 3.2.1.4. On compare la méthode DyCF, paramétrée avec différents degrés d, à la méthode par KDE, paramétrée avec différentes tailles de fenêtres. La méthode par KDE nécessite également de fixer un seuil sur le score, mais rappelons que ce paramètre n'affecte pas l'évaluation réalisée par WOLF-Eval.

Rappelons qu'on utilise 5 jeux de données différents :

— 3 jeux de données constitués d'une gaussienne, en dimensions 2, 3 et 4 ;
— 2 jeux de données constitués de huit gaussiennes, en dimensions 2 et 3.

La complexité de DyCF est comparée à celle de la méthode par KDE en mesurant et restituant le temps de traitement, comprenant la mise à jour des modèles et l'évaluation.

En ce qui concerne l'occupation mémoire, la méthode par KDE nécessite de l'espace mémoire pour :

— $W \times p$ flottants correspondant aux instances passées,
— la matrice $p \times p$ de largeur de bande qui, dans le cas de la règle de Scott, est une matrice diagonale, contenant donc seulement p éléments d'intérêt,
— la nouvelle instance à évaluer (p flottants) et son score,

tandis que la méthode DyCF nécessite de pouvoir stocker :

— $\frac{s(d)(s(d)+1)}{2}$ éléments pour l'inverse de la matrice des moments, qui est symétrique, et selon la méthode utilisée pour l'incrémentation du modèle, il peut être nécessaire de stocker également la matrice des moments elle-même,
— un entier correspondant au nombre d'instances observées jusqu'ici,
— pour chaque nouvelle instance $\mathbf{x}$ à évaluer, sa représentation $\mathbf{v}_d(\mathbf{x})$ sur $s(d)$ flottants, et son score.

Ainsi, savoir quelle méthode est la moins exigeante en occupation mémoire dépend essentiellement de la taille de la fenêtre W pour la méthode par KDE et de $s(d)$ pour DyCF.

Évaluation avec une gaussienne Pour l'évaluation sur une gaussienne, étant donnée la simplicité de la distribution, $d = 2$ devrait donner les meilleures performances pour DyCF. A titre de comparaison, les degrés $d = 4$ et $d = 6$ sont également évalués.

Les Figures 5.4, 5.6, et 5.8 donnent les résultats de l'évaluation selon WOLF-Eval tandis que les Figures 5.5, 5.7 et 5.9 donnent les durées d'évaluation de l'ensemble des instances.

On remarque tout d'abord que, peu importe p et d, DyCF donne de meilleurs résultats en précision que la méthode par KDE. Dans le plus mauvais des cas, avec $d = 6$, les résultats sont à peu près équivalents à la méthode par KDE avec $W = 1000$.

En ce qui concerne la durée d'évaluation en revanche, la complexité en $s_p(d)$ apparaît clairement lorsque d et p augmentent. Pour $p = 2$, on observe sur la Figure 5.5 que DyCF est plus performante même pour $d = 6$. En revanche, ce n'est plus le cas pour $p \geq 3$, comme en témoignent les Figures 5.7 et 5.9. DyCF avec $d = 2$ reste plus performante que la méthode par KDE pour les différentes tailles de fenêtre évaluées et au moins tant que $p \leq 4$, tandis qu'avec $d = 4$, DyCF est comparable à la méthode par KDE avec $W = 500$ et $W = 1000$ pour $p = 4$.

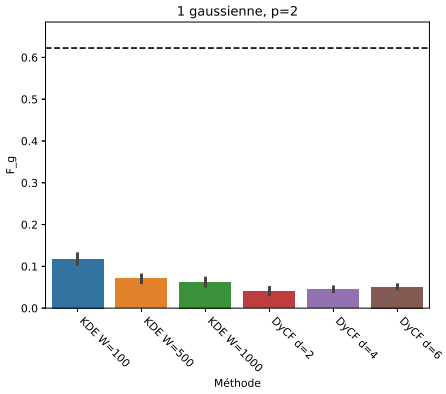

FIGURE 5.4 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 2$.

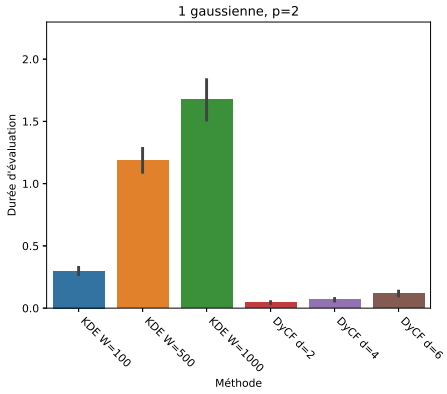

FIGURE 5.5 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 2$.

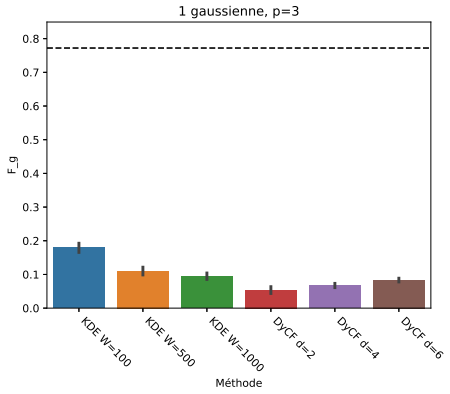

FIGURE 5.6 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 3$.

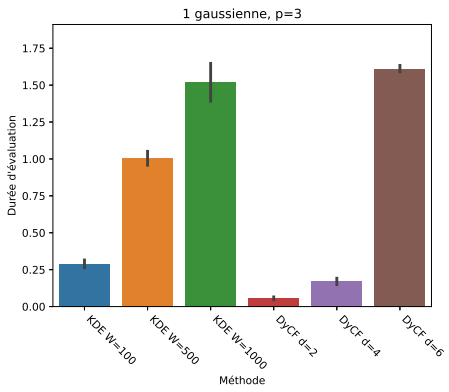

FIGURE 5.7 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 3$.

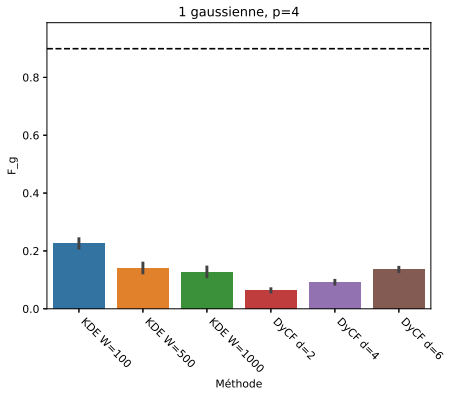

FIGURE 5.8 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 4$.

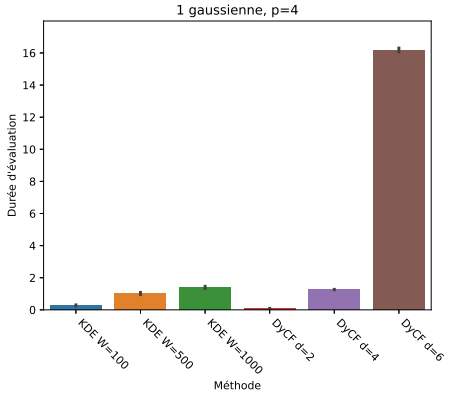

FIGURE 5.9 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec une gaussienne et $p = 4$.

Évaluation avec un mélange de huit gaussiennes Pour l'évaluation avec huit gaussiennes, la forme du support étant plus complexe, on évalue les valeurs 4, 6 et 8 pour le paramètre d.

Les résultats obtenus avec WOLF-Eval sont restitués sur les Figures 5.10 et 5.12 et les durées de traitement sont présentées dans les Figures 5.11 et 5.13.

Les résultats en précision sont cette fois à peu près similaires entre la méthode par KDE et DyCF. On remarque également une amélioration de la précision pour $d = 8$ par rapport à $d = 4$.

Au niveau de la durée d'évaluation, les différents paramétrages de DyCF proposent de meilleures performances pour $p = 2$, et seul le modèle avec $d = 8$ est

plus lent que le modèle par KDE avec $W = 100$. En revanche, pour $p = 3$, les performances sont plus équilibrées ; la méthode la plus performante est DyCF avec $d = 4$ suivie de la méthode par KDE avec $W = 100$. Les autres paramétrages sont à peu près équivalents, avec seulement le modèle de DyCF pour $d = 8$ qui apparaît bien plus lent que les autres.

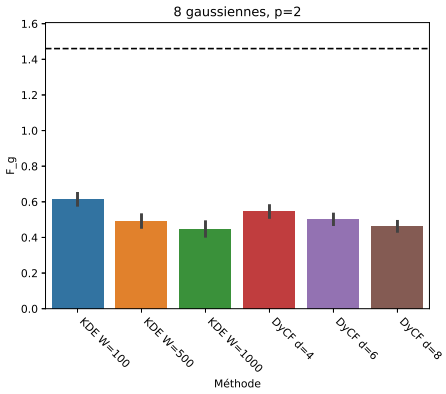

FIGURE 5.10 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies statistiques avec huit gaussiennes et $p = 2$.

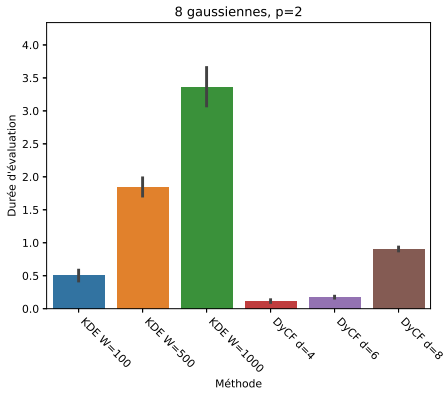

FIGURE 5.11 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec huit gaussiennes et $p = 2$.

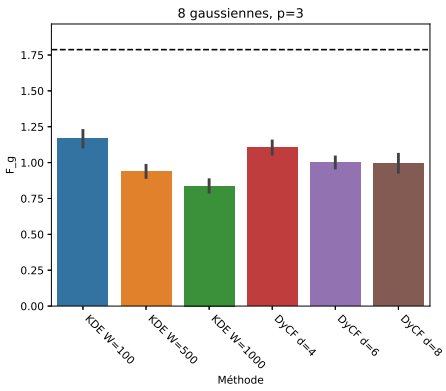

FIGURE 5.12 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec huit gaussiennes et $p = 3$.

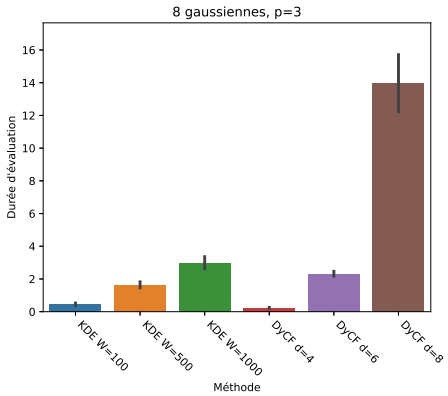

FIGURE 5.13 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLD-KDE pour les anomalies statistiques avec huit gaussiennes et $p = 3$.

Bilan de l'évaluation De manière générale, les tests réalisés ici ont démontré que WOLF-DyCF était plus performant que WOLF-KDE en ce qui concerne la définition statistique de l'anomalie.

En revanche, si ces tests ont montré une certaine stabilité sur la précision avec

le degré d, ils ont également mis en avant la forte croissance de la complexité, dépendante de p et d. Celle-ci est tout de même acceptable pour $p < 4$ et $d < 8$, ce qui est dans tous les cas imposé par les limites en capacité mémoire des équipements.

5.2.3 Définition basée distance

En maintenant l'hypothèse selon laquelle la FC a un comportement similaire à la densité de probabilité, on suppose que l'aire sous la FC permet également de suivre l'évolution du nombre d'instances. Cependant, contrairement à une KDE, et en particulier avec le noyau de Epanechnikov utilisé tout au long du Chapitre 3, où il est possible d'intégrer directement l'estimation de la densité, il est plus difficile d'intégrer la FCE, décrite comme l'inverse d'un polynôme.

Notons que, si la FC était une estimation de densité, il faudrait multiplier l'intégrale par le nombre d'instances dans le jeu de données pour retrouver le nombre de voisins. Cependant, puisque le nombre d'instances encodées par le modèle évolue et qu'on ne veut pas faire évoluer le seuil, on préfère n'étudier que l'intégrale sous la FC, qui donne alors une estimation du *taux de voisins*.

Une approche possible pour calculer l'intégrale est de l'estimer à partir d'échantillons. En particulier, dans les tests de ce chapitre, la méthode de Monte-Carlo est utilisée. Celle-ci repose sur un échantillonnage aléatoire du domaine sur lequel est défini l'intégrale à estimer.

Précisément, soit f la fonction à intégrer sur le domaine $I = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_p, b_p]$ de volume $V = \prod_{i=1}^{p} (b_i - a_i)$, on souhaite donc estimer :

$$\int_I f(\mathbf{x}) d\mathbf{x} \,. \tag{5.14}$$

Or, soit X une variable aléatoire multi-dimensionnelle qui suit une distribution uniforme dans I, de pdf $p(\mathbf{x}) = \frac{1}{V}$ pour $\mathbf{x} \in I$, on a pour toute fonction g l'espérance :

$$E(g(X)) = \int_I g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \,, \tag{5.15}$$

et soient $(\mathbf{x}_i)_{1 \leq i \leq N_s}$ N_s échantillons de la variable X, on peut évaluer l'espérance de manière empirique avec la somme :

$$\mathbf{\Sigma}_{N_s} = \frac{1}{N_s} \sum_{i=1}^{N_s} g(\mathbf{x}_i) \,. \tag{5.16}$$

En posant $g = \frac{f}{p}$, on déduit des Équations 5.15 et 5.16 que :

$$\int_I f(\mathbf{x}) d\mathbf{x} = E(\frac{f}{p}(X)) \tag{5.17}$$

et

$$\mathbf{\Sigma}_{N_s} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} = \frac{V}{N_s} \sum_{i=1}^{N_s} f(\mathbf{x}_i) \,. \tag{5.18}$$

On peut ainsi obtenir une estimation de l'intégrale de $\Lambda_d^{\mu_N}$ sur tout hypercube de côté $2R$, ce qui permet d'estimer les anomalies de distance de la même manière que dans WOLF-KDE, en fixant un seuil sur l'estimation $\mathbf{\Sigma}_{N_s,R}(\mathbf{x}_i) = \int_{\mathbf{x}_i-R}^{\mathbf{x}_i+R} \Lambda_d^{\mu_N}(\mathbf{x})d\mathbf{x}$ pour une instance $\mathbf{x}_i$. On retient alors comme score d'anomalies, pour établir le classement des instances dans WOLF-Eval, $\frac{1}{1+S_{N_s,R}}$.

Choix de N_s pour l'estimation On a $\lim_{N_s\to\infty} \mathbf{\Sigma}_{N_s} = \int_I f(\mathbf{x})d\mathbf{x}$ et on peut déterminer un intervalle de confiance, à 95% par exemple, sur l'estimation de l'intégrale comme :

$$\left[\mathbf{\Sigma}_{N_s} - \frac{1.96\sqrt{v_{N_s}}}{\sqrt{N_s}}, \mathbf{\Sigma}_{N_s} + \frac{1.96\sqrt{v_{N_s}}}{\sqrt{N_s}}\right],$$

où v_{N_s} est la variance empirique de f/p définie par :

$$v_{N_s} = \frac{V^2}{N}\left(\sum_{i=1}^{N_s} f(\mathbf{x}_i)^2\right) - \mathbf{\Sigma}_{N_s}^2 . \tag{5.19}$$

On souhaite cependant évaluer l'impact de N_s sur la précision selon WOLF-Eval afin de fixer ce paramètre pour la comparaison de WOLF-DyCF à WOLF-KDE. Pour se faire, reprenons le jeu de données utilisé pour la comparaison de la méthode par KDE à la méthode par M-Tree dans la Section 3.2.2.4 et comparons les résultats obtenus pour différentes valeurs de N_s, pour différentes combinaisons des paramètres d et R. Les résultats selon WOLF-Eval sont restitués en Figures 5.14 (pour $R = 0.5$) et 5.16 (pour $R = 1$).

On retourne également les durées d'évaluation afin de voir la conséquence du choix de N_s sur le temps de traitement, qui nécessite pour chaque instance d'évaluer la FCE en N_s points supplémentaires. Ces résultats sont affichés sur les Figures 5.15 et 5.17.

On remarque que, quelles que soient les valeurs de d ou R, les trois valeurs de N_s comparées donnent des résultats quasiment identiques. Ceci est principalement dû au fait que l'intégration est estimée sur de faibles volumes où la FCE prend des valeurs très faibles, en particulier pour les anomalies qui ont pourtant un poids plus important dans l'évaluation par WOLF-Eval.

En revanche, la durée d'évolution augmente linéairement avec N_s, et sa valeur sera donc fixée à $N_s = 100$ pour la suite.

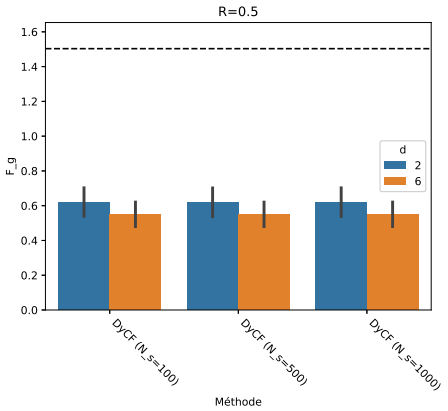

FIGURE 5.14 – Comparaison selon WOLF-Eval de WOLF-DyCF pour les anomalies de distance avec $R = 0.5$ pour différentes valeurs de N_s et $d \in \{2, 6\}$.

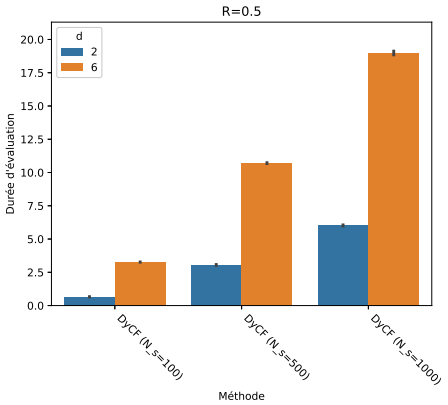

FIGURE 5.15 – Comparaison selon la durée d'évaluation de WOLF-DyCF pour les anomalies de distance avec $R = 0.5$ pour différentes valeurs de N_s et $d \in \{2, 6\}$.

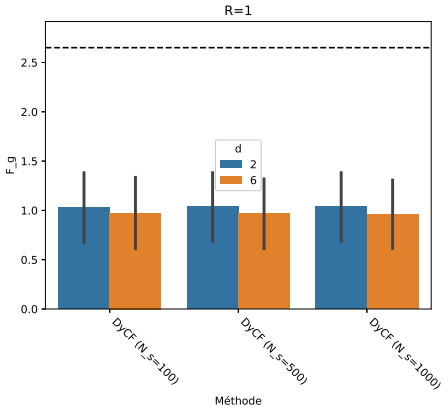

FIGURE 5.16 – Comparaison selon WOLF-Eval de WOLF-DyCF pour les anomalies de distance avec $R = 1$ pour différentes valeurs de N_s et $d \in \{2, 6\}$.

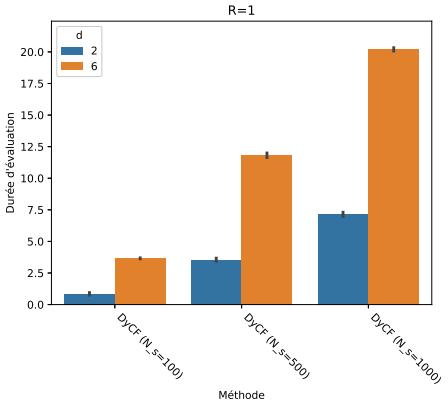

FIGURE 5.17 – Comparaison selon la durée d'évaluation de WOLF-DyCF pour les anomalies de distance avec $R = 1$ pour différentes valeurs de N_s et $d \in \{2, 6\}$.

***Comparaison de* WOLF-DyCF *et* WOLF-KDE** Le protocole expérimental et les paramètres choisis pour la méthode par KDE sont les mêmes que dans la Section 3.2.2.4. Pour DyCF, on prend différentes valeurs de d : 2, 4 et 6.

Les Figures 5.18 à 5.23 présentent les résultats obtenus pour $R \in \{0.5, 1, 2\}$ au niveau de la précision selon WOLF-Eval et de la durée d'évaluation. On peut y observer assez clairement que les résultats en précision sont à peu près équivalents pour WOLF-DyCF et WOLF-KDE. Néanmoins, pour $R = 2$, la Figure 5.22

témoigne d'une plus grande variance des résultats en moyenne pour les différents paramétrages de DyCF par rapport à ceux de l'approche par KDE.

Les performances en durée d'évaluation sont en revanche plus hétérogènes et dépendent grandement des paramètres W et d des deux méthodes. En observant que la précision est similaire quelles que soient les valeurs prises pour ces paramètres, il parait intéressant de comparer les performances en durée d'évaluation des modèles les plus légers, à savoir le modèle associé au paramètre $W = 100$ pour la méthode par KDE et au paramètre $d = 2$ pour la méthode reposant sur la FCE. Les deux modèles ont une durée d'évaluation similaire, avec un léger avantage pour l'approche par KDE.

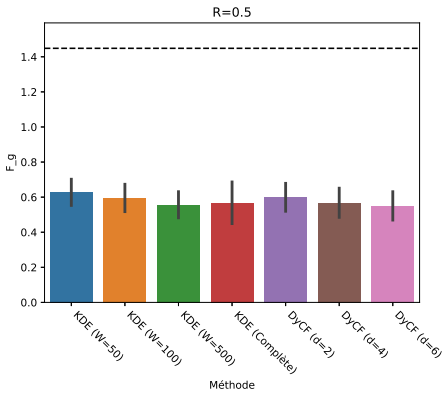

FIGURE 5.18 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 0.5$.

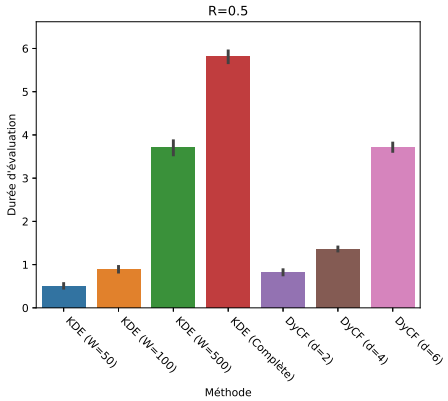

FIGURE 5.19 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 0.5$.

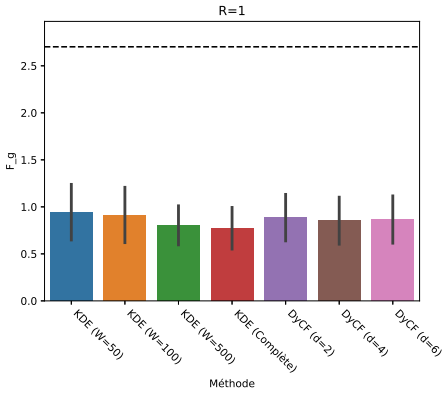

FIGURE 5.20 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 1$.

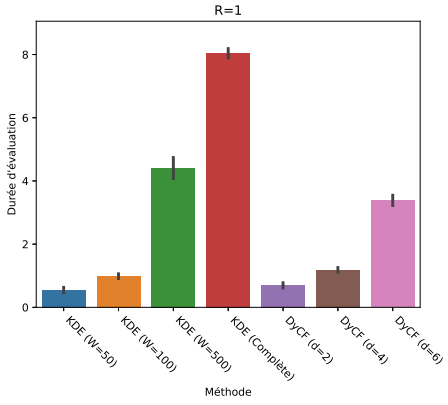

FIGURE 5.21 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 1$.

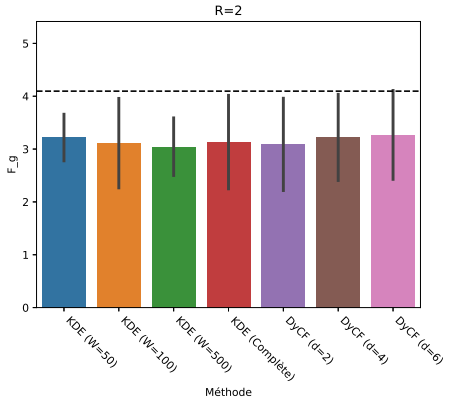

FIGURE 5.22 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 2$.

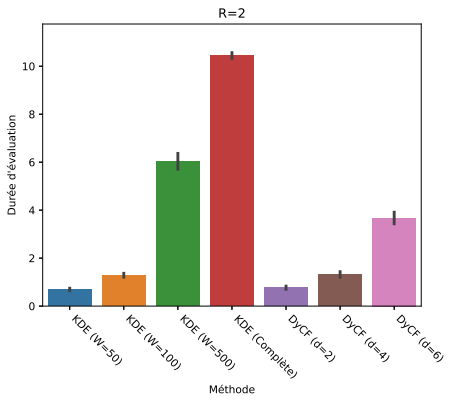

FIGURE 5.23 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance avec $R = 2$.

Bilan de l'évaluation Pour la définition de distance, les résultats obtenus montrent que WOLF-DyCF a des performances très similaires à WOLF-KDE, ce qui semble valider l'hypothèse selon laquelle l'intégrale de la FCE sur un voisinage a un comportement similaire au dénombrement des instances dans ce voisinage.

5.2.4 Définition basée densité locale

En maintenant l'hypothèse que l'intégrale de $\Lambda_d^{\mu_N}$ dans un voisinage d'une instance a un comportement similaire au dénombrement des instances dans ce voisinage, on peut reprendre le calcul du MDEF utilisé dans WOLF-KDE et décrit en Section 3.2.3.3 afin d'approximer son comportement avec DyCF au sein de WOLF-DyCF.

On choisit d'estimer l'intégrale avec la méthode de Monte-Carlo, comme pour les anomalies de distance, et on conserve $N_s = 100$.

***Comparaison de* WOLF-DyCF *et* WOLF-KDE** Le protocole expérimental suivi, ainsi que les paramètres choisis pour la méthode par KDE, sont les mêmes que dans la Section 3.2.3.4. Les valeurs de d retenues pour DyCF sont les mêmes que pour la définition de distance, à savoir 2, 4 et 6.

Les résultats obtenus sont présentés dans les Figures 5.24 à 5.29. Alors qu'il avait été observé en Section 3.2.3.4 que la précision variait peu pour la méthode par KDE en faisant varier W, on remarque sur les Figures 5.24, 5.26 et 5.28 que celle-ci diminue quand d augmente pour la méthode basée sur la FCE. Cette observation est problématique puisqu'elle témoigne de l'importance de convenablement fixer d pour les anomalies de densité locale, ce qui n'était pas le cas pour les définitions précédentes.

Les durées d'évaluation pour les différentes valeurs de R, décrites dans les Figures 5.25, 5.27 et 5.29, montrent des résultats similaires à ce qui avait été observé pour les anomalies de distance, avec une augmentation due à la multiplication du nombre de requêtes par 2^{pm}.

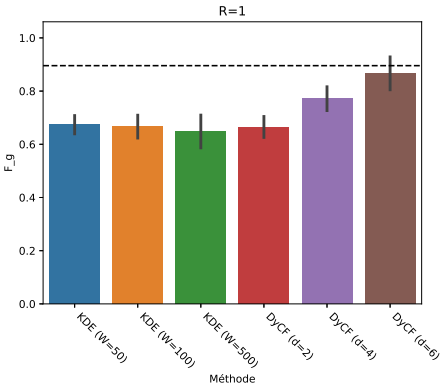

FIGURE 5.24 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 1$.

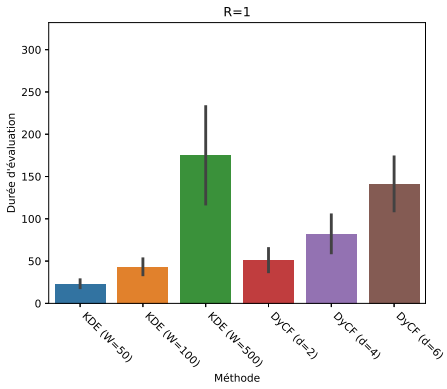

FIGURE 5.25 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 1$.

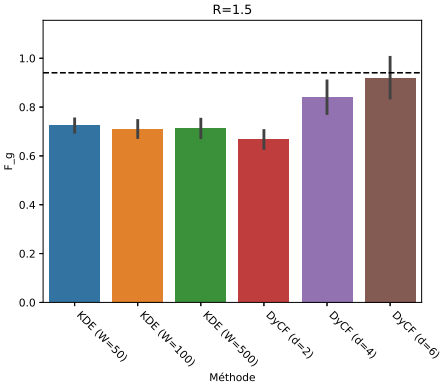

FIGURE 5.26 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 1.5$.

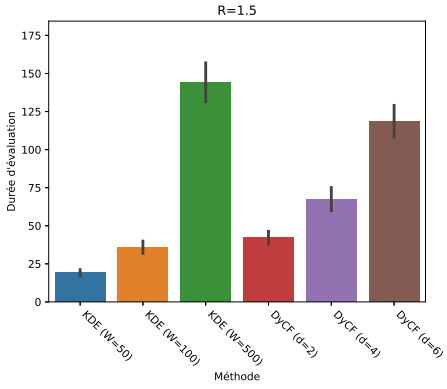

FIGURE 5.27 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 1.5$.

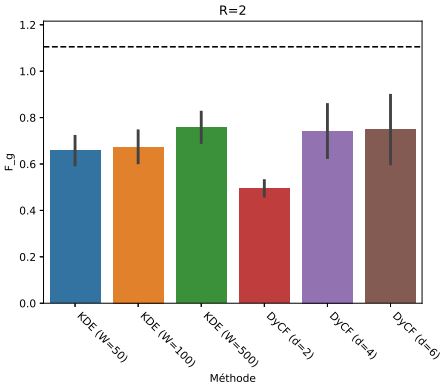

FIGURE 5.28 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 2$.

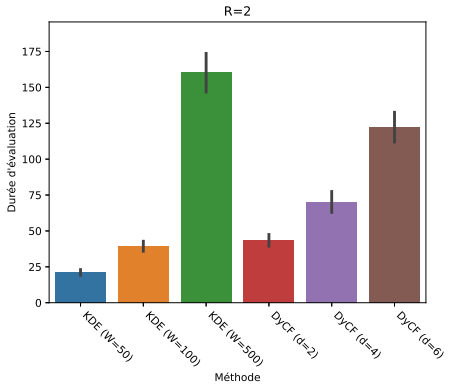

FIGURE 5.29 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale avec $R = 2$.

Bilan de l'évaluation La comparaison entre WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale montre une première faiblesse de WOLF-DyCF avec une plus forte dépendance au choix du paramètre d. En revanche, on peut noter que, pour d convenablement fixé, les résultats de WOLF-DyCF sont au moins aussi bons que ceux de WOLF-KDE. De plus, tout comme W, le paramètre d peut être fixé en fonction des capacités mémoires disponibles.

5.2.5 Évaluation sur un jeu industriel

Nous avons présenté, dans la Section 1.5.2 du Chapitre 1, un cas d'application des travaux de cette thèse. Celui-ci concerne un ensemble de convoyeurs et est associé à un jeu de données contenant des mesures pour deux grandeurs : la vitesse du tapis et l'intensité du moteur.

Dans cette sous-section, WOLF-DyCF et WOLF-KDE sont comparés selon ce jeu de données industriel pour étudier l'applicabilité de ces approches à un cas réel.

Cependant, ces données n'étant pas labellisées, il est nécessaire de s'appuyer sur WOLF-Eval pour évaluer les méthodes. Aussi, puisque le jeu de données n'est pas associé à une distribution statistique spécifique, il n'est pas possible d'appliquer WOLF-Eval à la définition statistique. Seules les méthodes basées distance et densité locale sont donc comparées.

Le jeu de données, contenant 21927 instances sur deux variables, est affiché en Section 1.5.2. Contrairement aux cas précédents, la temporalité a une importance puisque le convoyeur change d'état au cours du temps. WOLF-Eval est néanmoins utilisé de la même manière que précédemment ; les classements théoriques sont calculés en tenant compte de l'ensemble du jeu de données tandis que les classements issus des méthodes sont générés à partir du score calculé en ligne.

Puisqu'il n'y a plus d'aspect aléatoire dans le jeu de données, on ne reproduit plus l'expérience plusieurs fois et il n'y a plus de variance dans le résultat.

Définition de distance Pour l'évaluation des méthodes basées distance, on teste trois valeurs de R : 0.05, 0.1 et 0.2. Les valeurs sont plus faibles que dans les tests précédents à cause de la plus grande proximité des instances. On conserve en revanche des tailles de fenêtre de 50, 100 et 500 pour WOLF-KDE et des degrés de 2, 4 et 6 pour WOLF-DyCF. Les résultats selon WOLF-Eval sont fournis dans les Figures 5.30, 5.32 et 5.34 et les résultats en temps de traitement dans les Figures 5.31, 5.33 et 5.35.

Que ce soit en précision ou en temps de traitement, WOLF-DyCF permet d'obtenir de meilleures performances que WOLF-KDE pour les valeurs de R et les paramètres testés.

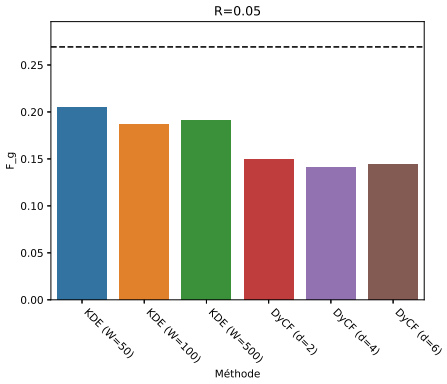

FIGURE 5.30 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.05$.

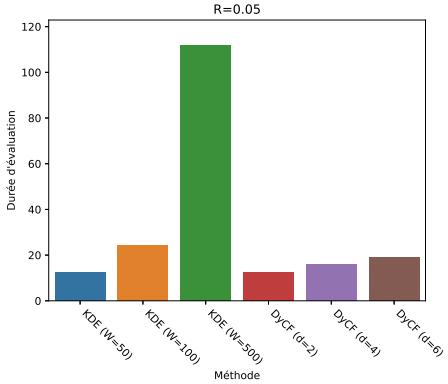

FIGURE 5.31 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.05$.

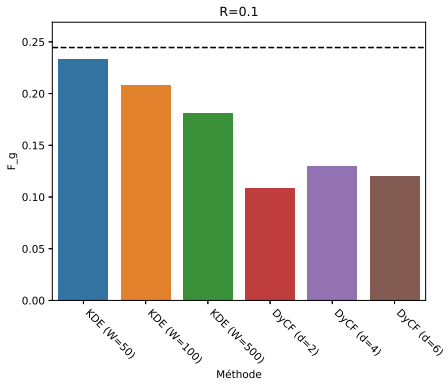

FIGURE 5.32 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.1$.

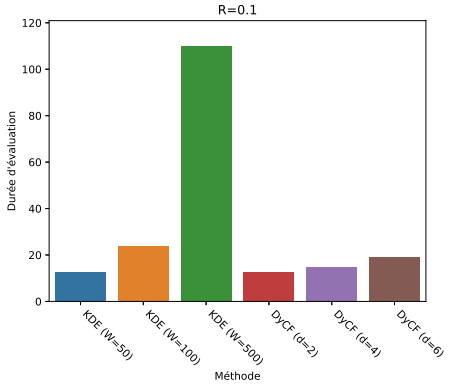

FIGURE 5.33 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.1$.

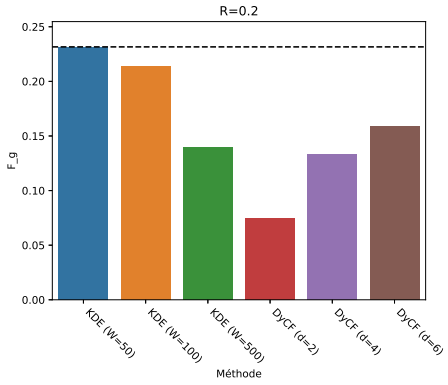

FIGURE 5.34 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.2$.

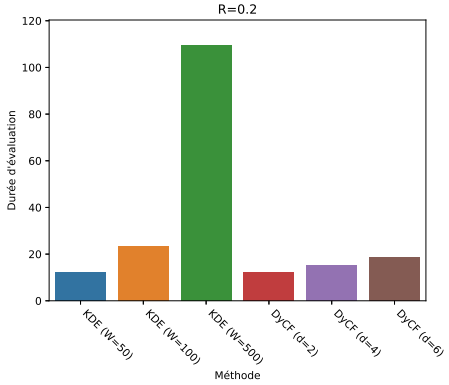

FIGURE 5.35 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de distance sur le jeu des convoyeurs avec $R = 0.2$.

Définition de densité locale Les valeurs du paramètre R testés pour l'évaluation des méthodes basées densité locales sont les suivantes : 0.1, 0.2 et 0.4. Les paramètres de WOLF-KDE et WOLF-DyCF sont les mêmes que pour la définition de densité. Les résultats selon WOLF-Eval sont fournis dans les Figures 5.36, 5.38 et 5.40 et les résultats en temps de traitement dans les Figures 5.37, 5.39 et 5.41.

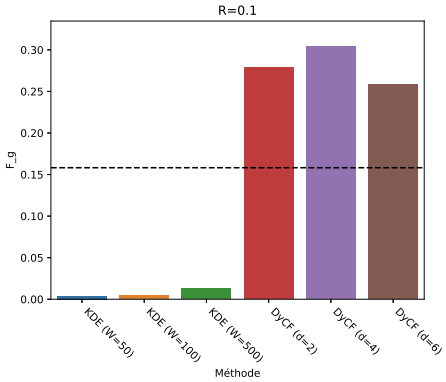

FIGURE 5.36 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.1$.

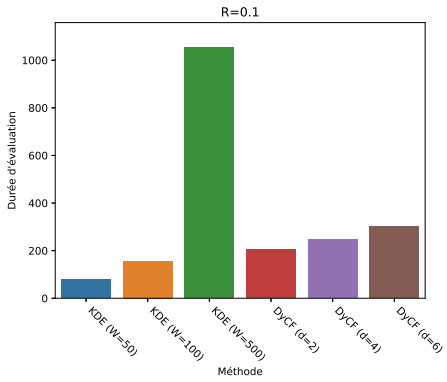

FIGURE 5.37 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.1$.

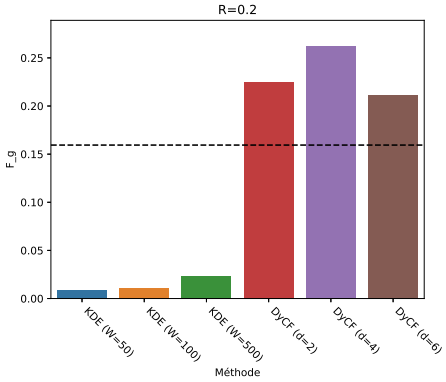

FIGURE 5.38 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.2$.

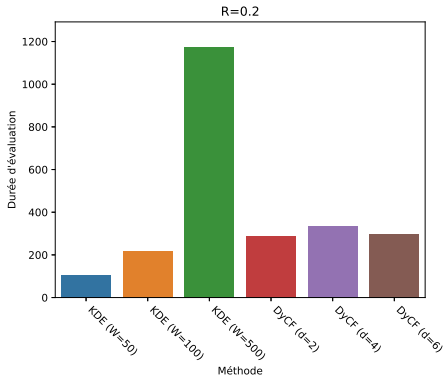

FIGURE 5.39 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.2$.

Les performances de WOLF-DyCF sont cette fois très mauvaises par rapport à WOLF-KDE. L'explication de ces performances remet en question la pertinence de l'utilisation de DyCF pour la détection des anomalies de densité locale ; en effet, lorsqu'une instance n'a pas de voisins, et parce que le MDEF est calculé comme le

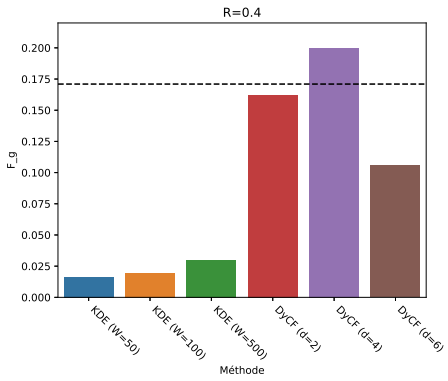

FIGURE 5.40 – Comparaison selon WOLF-Eval de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.4$.

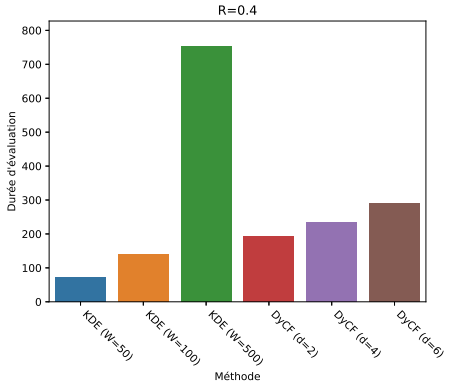

FIGURE 5.41 – Comparaison selon la durée d'évaluation de WOLF-DyCF et WOLF-KDE pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.4$.

rapport du nombre de voisins de l'instance sur le nombre de voisins moyen dans son voisinage, on renvoie un score d'anomalie maximal pour éviter la division par zéro.

Dans WOLF-KDE, du fait de l'utilisation du noyau de Epanechnikov qui est nul en dehors d'un hyper-cube autour du centre de noyau, la même approche est utilisée pour considérer comme anormales les instances dans des zones avec aucun voisin.

En revanche, pour WOLF-DyCF, le nombre de voisins est estimé à partir de la FCE qui n'est jamais nulle. Aussi, dans les zones de l'espace sans instances, la FCE prend des valeurs très faibles ; le nombre de voisins estimé d'une instance est alors proche de celui des autres instances de son voisinage, et l'instance est considérée comme normale, engendrant de mauvaises performances selon WOLF-Eval avec un écart de l'ordre de la taille du jeu de données dans les classements des instances anormales, qui ont un poids associé très élevé.

Une manière de résoudre ce problème serait de fixer une valeur nulle à l'estimation du nombre de voisins dans le cas où celle-ci serait suffisamment faible. La valeur seuil doit alors être dépendante du score de la FCE associé et du volume de l'intégration pour estimer le nombre de voisins. On choisit en particulier de prendre $d^{-3p/2}(2r)^p$, qui correspond à un score inférieur à $d^{-3p/2}$ pour la FCE puisque l'intégrale est calculée sur un hyper-cube de côté $2r$.

Les résultats ainsi obtenus sont présentés sur les Figures 5.42, 5.43 et 5.44. On remarque que la solution améliore grandement les performances sauf pour $d = 2$ avec $R = 0.1$ où les résultats n'ont pas été affectés.

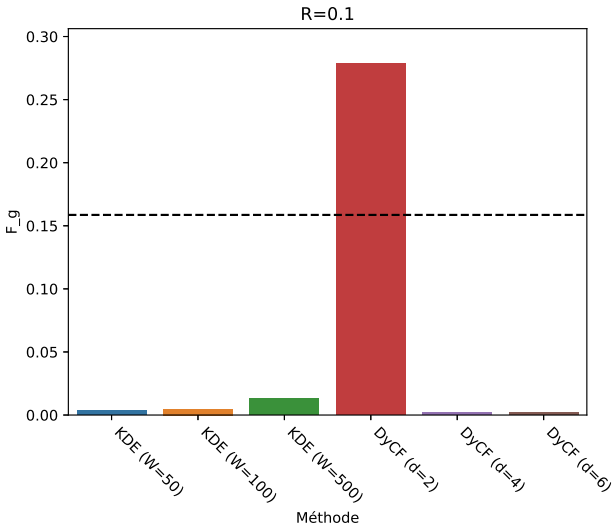

FIGURE 5.42 – Evaluation selon WOLF-Eval de WOLF-DyCF modifié pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.1$.

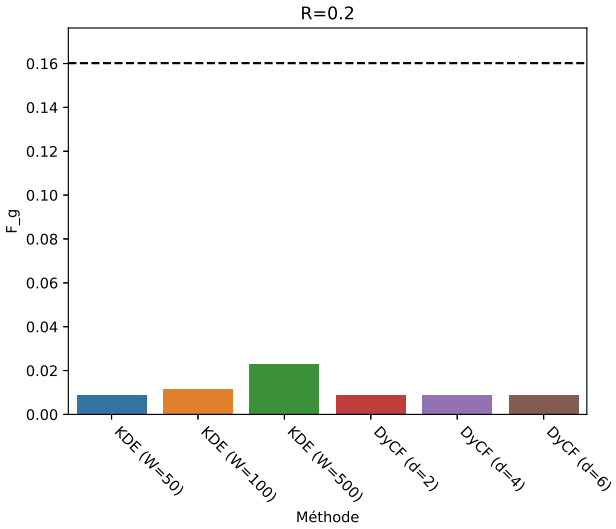

FIGURE 5.43 – Evaluation selon WOLF-Eval de WOLF-DyCF modifié pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.2$.

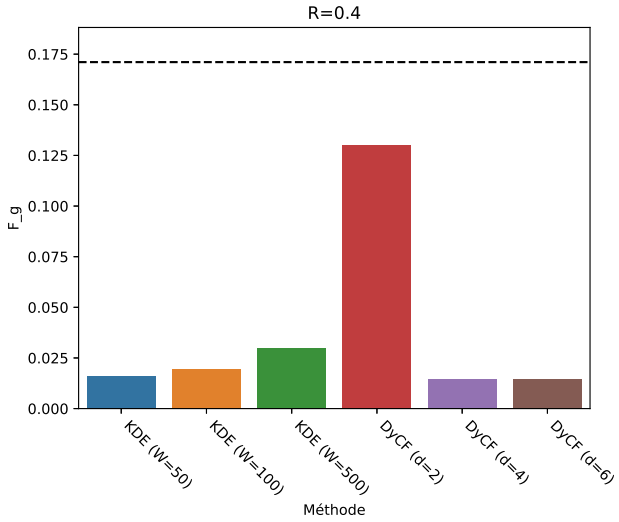

FIGURE 5.44 – Evaluation selon WOLF-Eval de WOLF-DyCF modifié pour les anomalies de densité locale sur le jeu des convoyeurs avec $R = 0.4$.

5.2.6 Observations générales

De manière générale, il semble que WOLF-DyCF puisse être une alternative convaincante à WOLF-KDE. Les performances selon la méthode d'évaluation WOLF-Eval sont à peu près équivalentes, et WOLF-DyCF fournit parfois même de meilleurs résultats, notamment sur le jeu industriel dans le cas de la définition de distance. Ainsi, l'hypothèse selon laquelle la FCE pouvait être considérée comme équivalente à la pdf dans son comportement semble vérifiée.

De plus, de manière similaire à W, le paramètre d peut être fixé selon les caractéristiques du problème, avec des possibilités fortement limitées par la croissance de $s_p(d)$, ce qui *facilite grandement le paramétrage de WOLF-DyCF*. Cette caractéristique est néanmoins nuancée par les variations des performances de WOLF-DyCF selon d dans le cas de la définition de densité locale, qui force un ajustement du paramètre.

Cependant, ces expérimentations ont également montré les limites de l'utilisation de la définition de densité locale avec une estimation reposant sur une fonction continue. Notons que l'utilisation du noyau gaussien à la place du noyau de Epanechnikov pour WOLF-KDE aurait eu les mêmes conséquences.

Enfin, les durées d'évaluation des méthodes au sein de WOLF-DyCF sont également intéressantes en comparaison à WOLF-KDE, principalement grâce à la stabilité de l'estimation de l'intégrale en fonction de N_s, liée au fait que les domaines d'intégration, dépendants de R, sont faibles, et qu'on s'intéresse principalement aux résultats dans les zones anormales où la FCE prend des valeurs proches de 0.

5.3 Limites et discussions

Cette dernière section discute de limites et possibles améliorations de la méthode basée sur la FCE proposée dans ce chapitre. De premiers tests sont réalisés et ouvrent sur de futurs travaux.

5.3.1 Retrouver le nombre d'instances

Il est difficile de fixer le seuil sur le score obtenu pour DyCF appliqué aux anomalies statistiques et de distance. Dans le cas des anomalies de distance en particulier, alors que le score est intuitif dans sa version exacte car dépendant du nombre de voisins, qui est un entier, l'interprétabilité du score est réduite avec DyCF étant donné que la FCE n'est pas une densité de probabilité ; son intégrale sur $\mathbb{R}^p$ ne vaut pas 1, et on ne peut donc pas la ramener à un dénombrement.

Une solution serait d'étudier le rapport entre l'estimation de l'intégrale sous la FCE, qui a tout de même un comportement similaire au dénombrement des instances dans le voisinage, et le dénombrement exact. Dans le cas où une relation apparaîtrait, on pourrait apporter une intuition derrière le choix du seuil.

Une première expérience dans ce sens, semblant valider empiriquement notre hypothèse, est décrite ci-après ; on choisit une distribution gaussienne en deux dimensions, centrée et réduite, et 1000 échantillons qui en sont tirés, correspondant à un ensemble d'entraînement. On prend ensuite un ensemble de test composé de 1000 nouveaux échantillons uniformément distribués dans $[-10, 10]^2$, et on calcule pour chacun le nombre exact de voisins à R qu'il possède dans l'ensemble d'entraînement décrit par la gaussienne, où R est calculé selon la méthode décrite dans la Section 4.2.3.1 sur l'ensemble d'entraînement. Pour le score par la DyCF, on entraîne d'abord le modèle sur l'ensemble d'entraînement et on estime ensuite le score de chaque instance de l'ensemble test correspondant à la loi uniforme.

Une fois le score estimé par DyCF et le nombre exact de voisins calculés pour plusieurs valeurs de d, on obtient le graphe de la Figure 5.45 affichant les combinaisons de valeurs obtenues pour chaque instance de l'ensemble test. On remarque facilement une relation qui semble linéaire entre l'estimation de l'intégrale sous la FCE et le nombre exact de voisins. Une droite est donc tracée par régression. Le graphe fourni le coefficient a de la relation $y = ax$, où y est le nombre exact de voisin et x la valeur estimée par DyCF, ainsi que le coefficient de détermination r^2, nombre compris entre 0 et 1 mesurant la qualité de la régression et calculé comme le rapport entre la somme des carrés des écarts des valeurs prédites et exactes à la moyenne. Les deux coefficients sont arrondis au millième.

Ce coefficient est clairement croissant en d, et des tests supplémentaires pourraient être réalisés pour étudier la relation exacte, ainsi qu'une éventuelle dépendance à la dimension p.

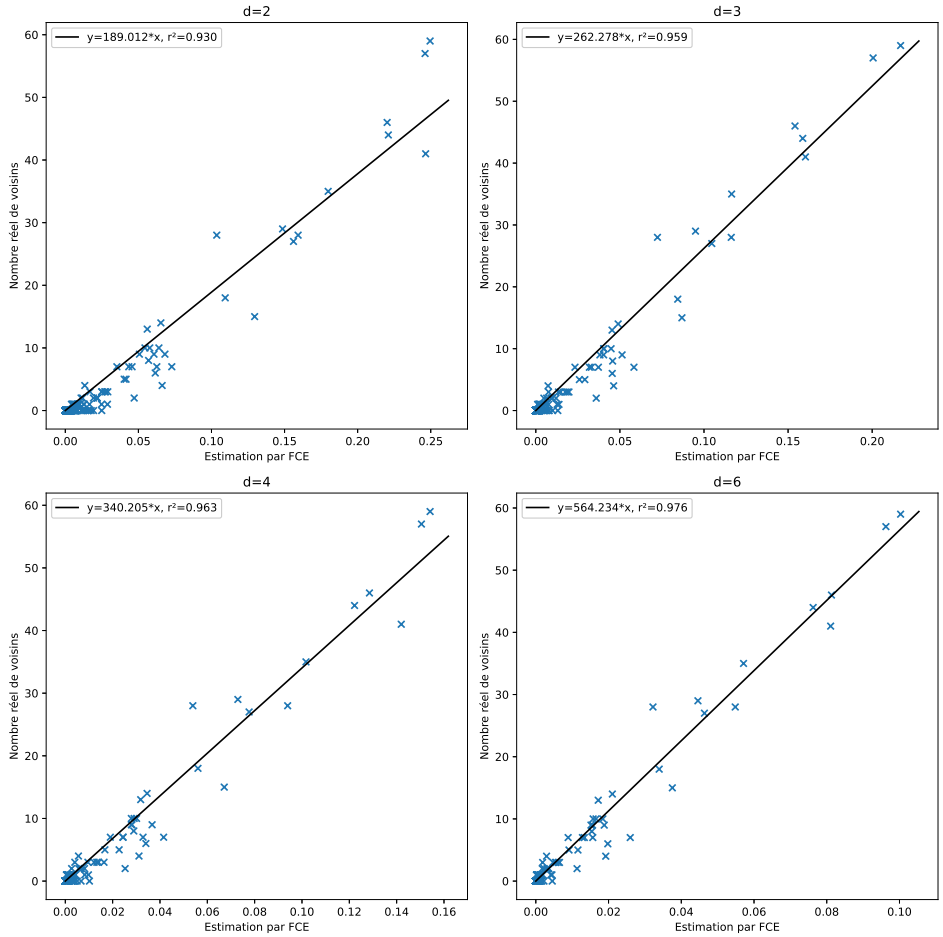

FIGURE 5.45 – Graphique représentant la relation linéaire entre (1) l'estimation du voisinage par intégration de la FCE et (2) le nombre exact de voisins.

5.3.2 Instabilités et choix de la base

Bien que celles-ci n'aient pas été observées dans les expériences menées dans ce chapitre, des instabilités calculatoires peuvent apparaître avec la méthode basée sur la FCE. Précisément, alors que la matrice des moments devrait théoriquement être semi-définie positive, c'est-à-dire avec uniquement des valeurs propres positives, il arrive que des valeurs propres négatives, de valeurs absolues très faibles, apparaissent, notamment lorsque les données ne sont pas centrées.

Bien que les valeurs propres aient un faible ordre de grandeur, il faut noter que le score $Q_d^{\mu_N}$ peut être fortement impacté car il est calculé sur l'inverse de la matrice des moments. Aussi, la négativité de certaines valeurs propres cause des valeurs négatives très élevées pour le score, pourtant théoriquement strictement positif.

Le problème décrit ici peut être corrigé en centrant les données. Cependant, pour des flux de données, il n'est pas garanti que celles-ci restent centrées, et puisqu'on ne conserve pas les instances passées en mémoire, réaliser un nouvel apprentissage sur des données centrées causeraient une perte de l'apprentissage précédemment réalisé.

Théoriquement, l'utilisation d'une base différente pour les polynômes de degrés au plus d pourrait apporter de la stabilité au calcul. Aussi, réaliser une inversion de Pearson, c'est-à-dire en remplaçant les valeurs propres négatives par des valeurs propres nulles, retire complètement les instabilités, mais au prix de conséquences sur les courbes de niveau.

La Figure 5.46 présente les courbes de niveau pour différents degrés d et différentes bases des polynômes. Parmi elles, on retrouve la base des monômes mais également la base de Legendre, la base de Tchebychev de la première espèce, avec sa représentation trigonométrique ou non, et la base de Tchebychev de la seconde espèce.

Pour représenter ce problème, le jeu de données précédent est ensuite décentré et les courbes de niveau sont de nouveaux tracées (voir Figure 5.47). On ne remarque pas d'instabilités pour $d = 2$ ou $d = 4$, mais elles apparaissent très clairement à partir de $d = 6$. De plus, bien que les instabilités diffèrent selon la base choisie, elles sont présentes sur chacune d'entre elles.

En modifiant l'inversion de la matrice des moments par l'inversion de Pearson, on règle le problème d'instabilités mais en étirant les courbes de niveau vers le centre du repère, comme montré dans la Figure 5.48, ce qui n'est pas plus enviable puisqu'on risque alors de considérer un grand nombre d'instances théoriquement anormales comme des instances normales. On peut également remarquer que la transformation opère dès $d = 4$, ce qui signifie que des valeurs propres étaient déjà négatives mais n'avaient pas d'impact sur les courbes de niveau affichées pour une inversion normale (cf. Figure 5.47).

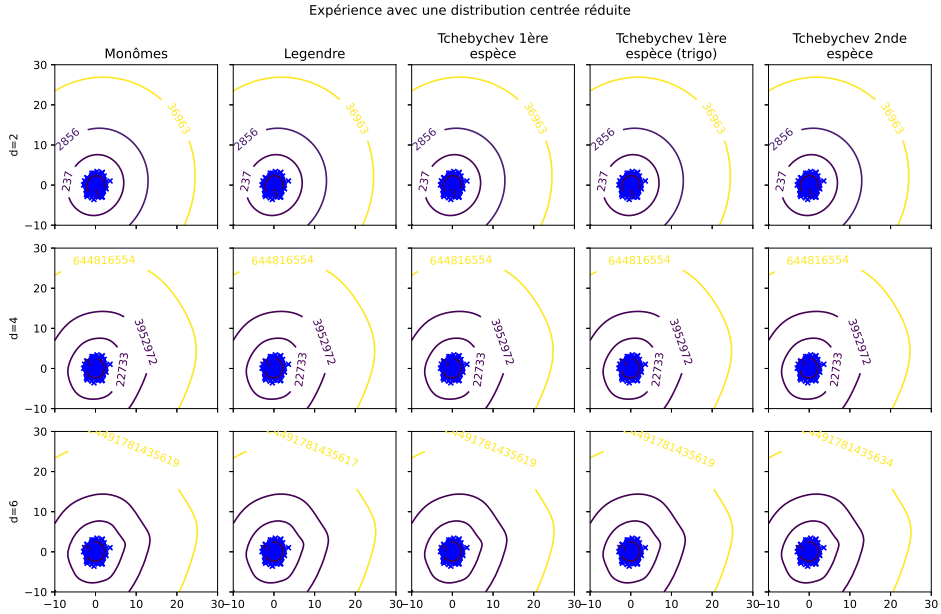

FIGURE 5.46 – Courbes de niveau obtenues pour différentes valeurs de d et différentes bases sur les données centrées.

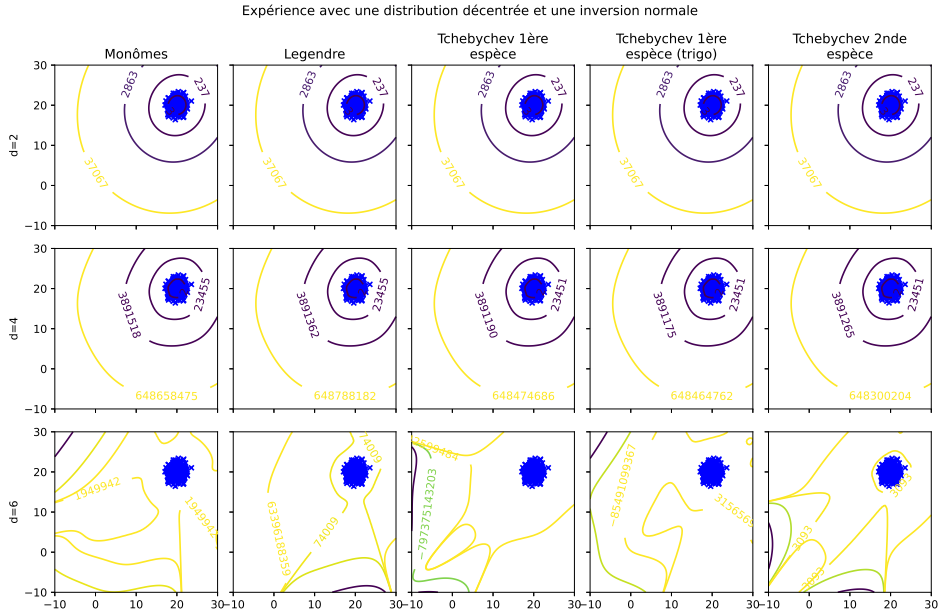

FIGURE 5.47 – Courbes de niveau obtenues avec une inversion normale pour différentes valeurs de d et différentes bases sur les données décentrées.

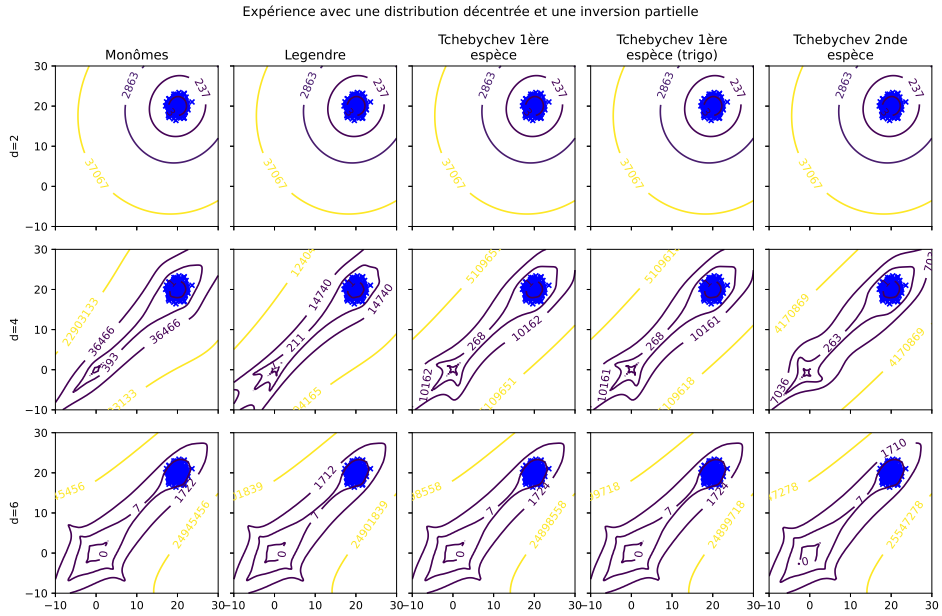

FIGURE 5.48 – Courbes de niveau obtenues avec une inversion de Pearson pour différentes valeurs de d et différentes bases sur les données décentrées.

5.3.3 Limites de l'hypothèse

Dans ce chapitre, nous faisons l'hypothèse que la FC peut être considérée comme une fonction de densité de probabilité. Le rapport entre la FC et la fonction de densité f a cependant été formellement caractérisée et on a :

$$\binom{p+d}{d}\Lambda_d^\mu(\mathbf{x}) \xrightarrow[d\to\infty]{} f(\mathbf{x})/\omega_E(\mathbf{x}) \tag{5.20}$$

où ω_E est la densité d'une mesure particulière liée à Ω et généralement inconnue.

Cependant, des travaux récents ont montré la convergence d'une fonction de Christoffel modifiée $\tilde{\Lambda}_d^\mu(\mathbf{x},\varepsilon)$ vers la densité de la distribution [Lasserre 2023]. Sous certaines hypothèses, il est démontré que :

$$\varepsilon^{-p}\tilde{\Lambda}_d^\mu(\mathbf{x},\varepsilon) \xrightarrow[d\to\infty]{} f(\zeta_\varepsilon) \approx f(\mathbf{x}) \tag{5.21}$$

pour $\varepsilon > 0$ suffisamment petit. Le remplacement de la FC par la FC modifiée n'a pas encore été expérimentée dans nos travaux mais devrait permettre d'améliorer encore les résultats de ce chapitre et de résoudre le problème mentionné en Section 5.3.1.

5.3.4 Question des dépendances temporelles

Dans le Chapitre 2, il a été établi qu'une méthode de détection d'anomalies adaptée aux WSNs devait traiter les dépendances temporelles au moins en considérant de larges plages temporelles à travers son modèle. Cependant, il faut différencier le

cas des KDE, qui reposent sur une fenêtre glissante avec un oubli des instances les plus anciennes, et le cas de DyCF, où la matrice des moments considère toutes les instances avec le même poids.

Il n'y a, à notre sens, pas une solution meilleure que l'autre et, selon le cas d'application, l'oubli des instances les plus anciennes peut être un avantage ou un inconvénient. Cependant, dans le cas où l'oubli serait souhaité, il est possible de l'intégrer à la construction incrémentale de la matrice des moments en assignant à chaque instance, ou plus précisément à chaque $\mathbf{v}(\mathbf{x}_i)^T\mathbf{v}(\mathbf{x}_i)$ dans la description de la matrice des moments empirique donnée par l'Équation 5.6, un poids correspondant à un facteur d'oubli exponentiel.

Néanmoins, il faut alors ajouter un paramètre correspondant à la vitesse d'oubli et on risque d'accélérer le déplacement de la distribution qui, lorsqu'elle n'est pas centrée, introduit des irrégularités dans le calcul.

5.3.5 Conclusion

Ce chapitre a étudié une nouvelle méthode de détection d'anomalies à intégrer dans WOLF. En comparaison, WOLF-DyCF donne des résultats assez similaires à WOLF-KDE, et ne repose pas sur des fenêtres glissantes. Cependant, la sensibilité des méthodes de WOLF-DyCF au paramétrage reste forte, notamment au choix de d, bien que celui-ci puisse être fixé selon les caractéristiques des équipements sur lesquels les modèles sont embarqués.

Chapitre 6

Proposition d'une méthode sans paramètres

Le chapitre précédent a présenté une nouvelle méthode pour la détection d'anomalies en ligne, DyCF, et son intégration au cadre opérationnel sous la forme de WOLF-DyCF. Bien que DyCF ait un paramétrage facilité par des propriétés de la FCE et des hypothèses sur la mesure associée aux données, il reste à fixer le paramètre d et on peut questionner les hypothèses réalisées. Dans ce sixième chapitre, les limites du paramétrage de DyCF dans un cadre général sont étudiées et une alternative sans paramètres nommée *DyCG* pour *Dynamical Christoffel Growth* est proposée.

Sommaire

6.1 DyCG : Suppression du paramétrage

Il a été mentionné et observé dans les chapitres précédents que le paramétrage des méthodes était un point bloquant dans un cadre non supervisé.

L'évaluation par WOLF-Eval proposée dans la Section 5.2 a montré une certaine stabilité de WOLF-DyCF selon le paramètre d. Cependant, dans un cadre plus général, cette stabilité des résultats n'est pas garantie.

Aussi, la question du seuil sur le score n'a pas été posée lors de ces tests du fait des spécificités de WOLF-Eval. En revanche, fixer un seuil reste nécessaire dans un cas appliqué pour assister l'opérateur du système étudié. Dans la Section 5.1.2, l'ECF a été présentée avec un seuil inspiré des travaux de [Vu 2020], en proposant notamment de prendre une fonction score $\mathbf{S}_{d,\gamma_C}$ avec un seuil naturel à 1 sur le score, et en fixant $C = 1$. On peut néanmoins se demander si ce seuil, dépendant de la valeur choisie pour C, est pertinent et adapté à n'importe quel jeu de données.

Ainsi, on peut discuter le choix des paramètres d et C. Il est alors intéressant de préciser que l'évolution de la FC par rapport au degré d a été théoriquement

caractérisée et permet de ne pas avoir à fixer ce paramètre. Cette caractérisation offre également un seuil sur le score obtenu.

En fixant une instance $\mathbf{x} \in \mathbb{R}^p$, l'évolution de $\Lambda_d^\mu(\mathbf{x})$ quand d croit dépend entièrement de l'appartenance de $\mathbf{x}$ au support de μ. Plus précisément, pour $\mathbf{x} \notin \Omega$, $Q_d^\mu(\mathbf{x}) = 1/\Lambda_d^\mu(x)$ suit une croissance exponentielle en d, ce qui n'est pas le cas pour $\mathbf{x} \in \Omega$.

La croissance exponentielle selon d pour $\mathbf{x}$ en dehors du support est qualifiée par le Théorème 1 :

Théorème 1 *([Lasserre 2022] Lemma 4.3.1 p.50) Soit μ une mesure borélienne positive, supportée par un ensemble compact $\Omega \subset \mathbb{R}^p$ de diamètre* $\mathrm{diam}(\Omega)$*, et soient $\mathbf{x} \notin \Omega$ et $\delta > 0$ tels que* $\mathrm{dist}(\mathbf{x}, \Omega) > \delta$*. Dans ce cas,*

$$Q_d^\mu(\mathbf{x}) \;\geq\; s_p(d) 2^{\frac{\delta d}{\delta + \mathrm{diam}(\Omega)} - 3} \left(\frac{p}{ed}\right)^p \exp\left(-\frac{p^2}{d}\right).$$

En parallèle, le Théorème 2 montre que $Q_d^\mu(\mathbf{x}) = 1/\Lambda_d^\mu(x)$ a une croissance au plus polynomiale en d pour les points à l'intérieur du support (avec $s_p(d) = \binom{p+d}{d}$ équivalent à d^p pour p fixé) :

Théorème 2 *([Lasserre 2022] Lemma 4.3.2 p.51) Soit μ une mesure borélienne positive, supportée par un ensemble compact $\Omega \subset \mathbb{R}^p$, adhérence d'un domaine borné U, et soient $\mathbf{x} \in U$ et $\delta > 0$ tels que* $\mathrm{dist}(\mathbf{x}, \partial U) \geq \delta$*. Dans ce cas,*

$$Q_d^\mu(\mathbf{x}) \;\leq\; s_p(d) \frac{C_p}{\delta^p} (1+p)^{-3},$$

où C_p ne dépend pas de d mais seulement de p fixé.

En s'appuyant sur les résultats asymptotiques des Théorèmes 1 et 2, une seconde méthode, nommée *DyCG* pour *Dynamical Christoffel Growth*, est conçue pour établir l'anormalité d'une instance en se basant sur deux modèles issus de DyCF avec des degrés d_{min} et d_{max} différents. De manière assez naturelle, on fixe $d_{min} = 2$, et d_{max} doit être choisi par rapport aux contraintes de mémoire ; à titre d'exemple, la valeur $d_{max} = 6$ donne une matrice des moments de taille 28×28 avec $p = 2$, et un total de 406 paramètres par symétrie de la matrice, contre 21 paramètres pour $d = 2$.

En utilisant $\mathbf{S}_{d,\gamma_C}(\mathbf{x}) = C d^{-3p/2} Q_d^{\mu_N}(\mathbf{x})$ comme fonction de score pour les modèles associés à d_{min} et d_{max}, et d'après l'Équation 5.10, on obtient que si $Q_d^{\mu_N}(\mathbf{x})$ suit une croissance au moins en $d^{3p/2}$ alors :

$$\mathbf{S}_{(d_{min},d_{max}),\gamma_C}(\mathbf{x}) = C \frac{\mathbf{S}_{d_{max},\gamma_1}(\mathbf{x}) - \mathbf{S}_{d_{min},\gamma_1}(\mathbf{x})}{d_{max} - d_{min}} \tag{6.1}$$

est positif. On retient donc $\mathbf{S}_{(d_{min},d_{max}),\gamma_C}$ comme fonction de score pour DyCG avec un seuil naturel à 0 quel que soit C. On peut alors fixer $C = 1$ sans conséquences sur le seuil et noter $\mathbf{S}_{(d_{min},d_{max})} := \mathbf{S}_{(d_{min},d_{max}),\gamma_1}$.

Pour illustrer cette nouvelle approche, on entraîne un modèle sur un nuage de points issu d'une distribution uniforme bivariée puis on évalue Q_d^μ pour différents d et pour trois instances différentes : une première au centre du support, une seconde en bordure à l'intérieur et une troisième à l'extérieur. La Figure 6.1 montre l'évolution de Q_d^μ pour ces différents points en fonction de d ; on remarque une croissance presque linéaire à l'intérieur du support et exponentielle à l'extérieur. La Figure 6.2 suit le même principe en traçant les courbes pour $\mathbf{S}_{d,\gamma_1}$; cette fois, on observe une décroissance en fonction de d pour les instances à l'intérieur du support et une croissance toujours exponentielle à l'extérieur.

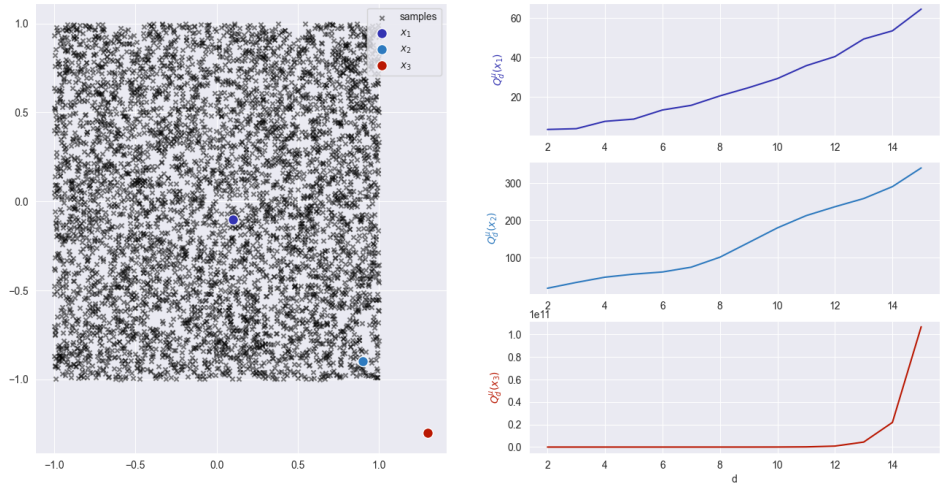

FIGURE 6.1 – Évolution de Q_d^μ pour différentes instances positionnées par rapport au support d'une distribution uniforme bivariée

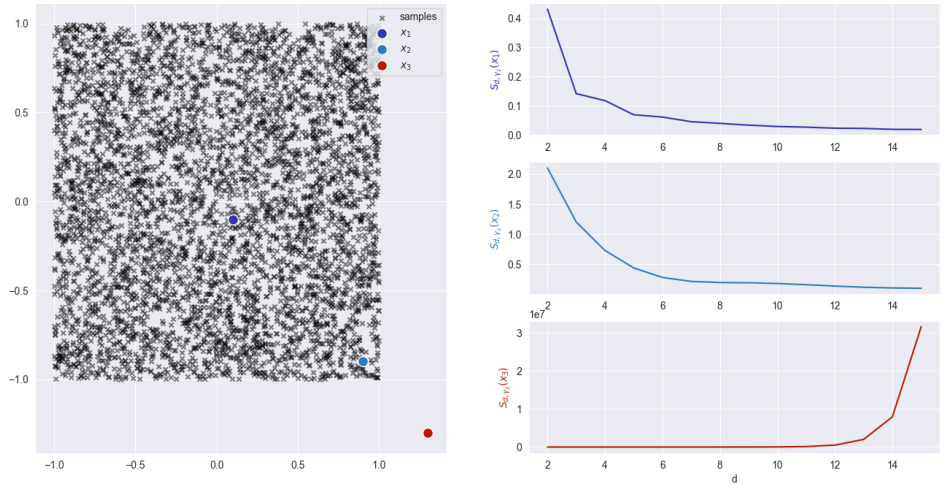

FIGURE 6.2 – Évolution de $\mathbf{S}_{d,\gamma_1}$ pour différentes instances positionnées par rapport au support d'une distribution uniforme bivariée

DyCG est donc une méthode qui atteint l'objectif de ne pas nécessiter de paramétrage spécifique, en dehors de d_{max} qui est fixé selon les caractéristiques du système embarquant les modèles. En effet, $s_p(d)$ a une croissance quasi-exponentielle en d et p, ce qui limite le choix de d_{max} sur des appareils de faible capacité mémoire.

Cependant, bien que DyCG puisse également être considérée comme une méthode statistique, l'hypothèse selon laquelle le score de DyCF suit un comportement similaire à la pdf ne tient plus pour DyCG. De ce fait, cette seconde méthode n'est pas évaluée au sein de WOLF dans cette thèse.

6.2 Performances de DyCG et limites de DyCF

Comme dans les expériences réalisées autour de SuMeLI, il est possible d'estimer le meilleur paramètre C pour DyCF, pour un degré d fixé, en étudiant les courbes ROC et Rappel-Précision sur un jeu de données labellisé.

Reprenons les jeux de tests présentés dans le Chapitre 4, à savoir le jeu des deux disques et celui des deux lunes avec leurs déclinaisons aléatoires et successives. Pour $d = 2$, $d = 4$ et $d = 6$, la meilleure valeur de C selon le ROC est déduite de la courbe ROC comme celle qui maximise $R \times (1 - FPR)$, avec R le rappel et FPR le taux de faux positifs, et la meilleure valeur selon le Rappel-Précision est déduite de la courbe Rappel-Précision comme celle qui maximise $P \times R$, avec P la précision. On déduit alors une valeur globale pour C comme celle maximisant le produit $R \times (1 - FPR) \times P \times R$.

Cependant, le produit $R \times (1 - FPR) \times P \times R$, que l'on notera ω, n'étant défini que pour les seuils associés à des anomalies du jeu de données, présentes en faibles proportions, le choix de la valeur optimale C est peu précis. On le calcule donc par une moyenne pondérée des C autour de la valeur optimale observée ; supposons les C_i indexés par le classement des anomalies, et ω_i le produit des métriques associées à C_i, plutôt que de choisir $C = C_k$ où k est tel que $\forall i, \omega_k > \omega_i$, on prend $C = \frac{\omega_{k-1}C_{k-1}+\omega_k C_k+\omega_{k+1}C_{k+1}}{\omega_{k-1}+\omega_k+\omega_{k+1}}$.

La Figure 6.3 illustre ce procédé pour la variante aléatoire du jeu des deux disques avec $d = 2$. Les courbes ROC et Rappel-Précision sont tracées sur les deux premiers graphes, accompagnées des produits $R \times (1 - FPR)$ en fonction du FPR et $P \times R$ en fonction de R associés. Le troisième graphe représente l'évolution des différents produits en fonction de C, et on s'intéresse en particulier au pic de la courbe verte qui correspond à $R \times (1 - FPR) \times P \times R$. En arrondissant au centième, C est alors encadré par $C_{k-1} = 0.13$ et $C_{k+1} = 0.15$, et on estime $C = 0.14$ selon la méthode décrite précédemment.

Les différentes valeurs de C obtenues à travers cette méthode et pour les combinaisons des quatre jeux de données et des trois valeurs de d sont fournies dans le Tableau 6.1.

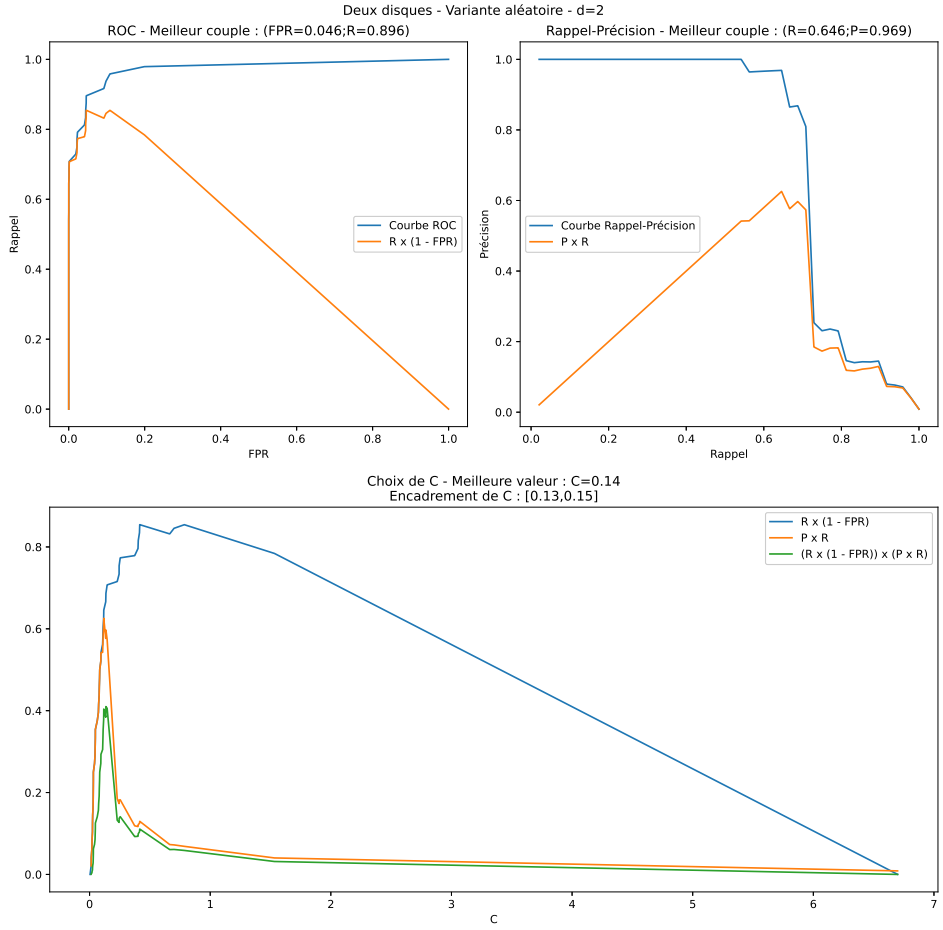

FIGURE 6.3 – Illustration du procédé de choix d'une valeur optimale pour C avec $d = 2$ et la variante aléatoire du jeu des deux disques.

	Deux disques		Deux lunes	
	Aléatoire	Successive	Aléatoire	Successive
$d = 2$	0.14	0.06	0.29	0.30
$d = 4$	0.30	0.10	0.57	0.10
$d = 6$	0.57	0.10	0.46	0.08

TABLE 6.1 – Valeurs de C optimales (arrondies au centième) estimées à partir des tests sur les courbes ROC et Rappel-Précision.

A partir des valeurs optimales de C obtenues, on trace le rappel, la précision et la f-mesure et on les compare aux performances pour $C = 1$. La Figure 6.4 donne les résultats obtenus. Les performances de DyCG, avec $d_{max} = 6$, sont également fournies et comparées.

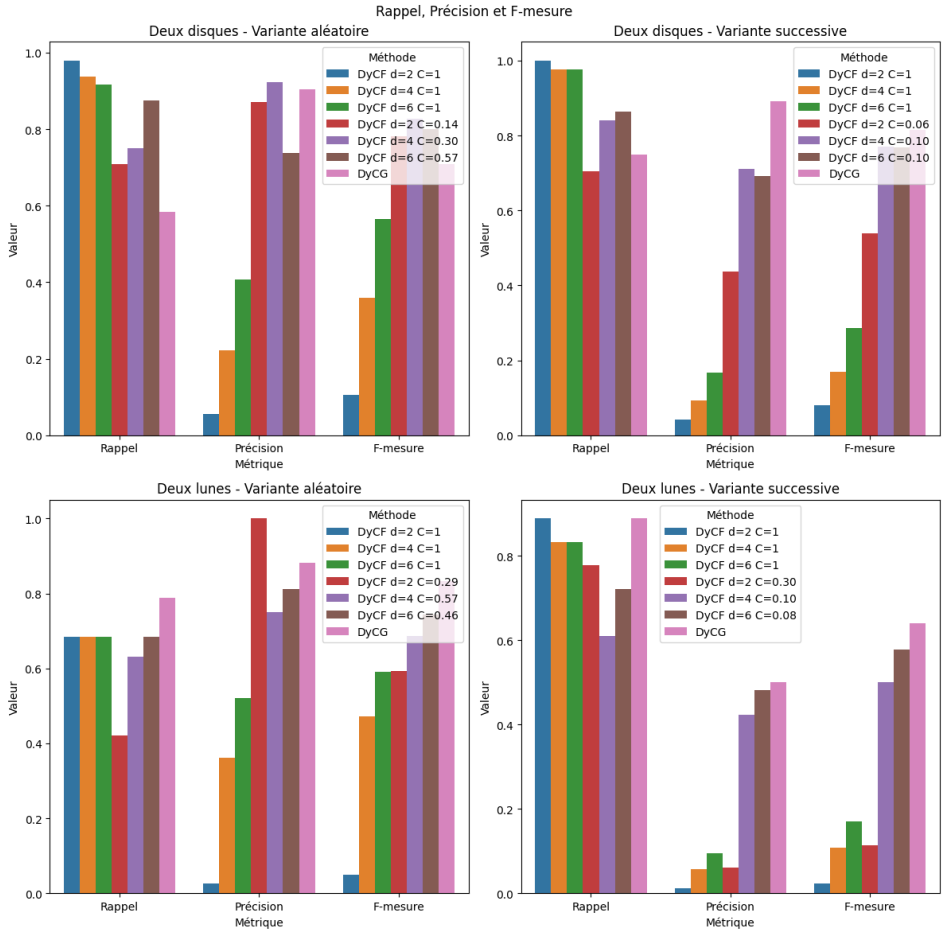

FIGURE 6.4 – Comparaison des performances en rappel, précision et f-mesure pour différents paramétrages de d et C.

L'importance du choix des paramètres d et C est indéniable sur les quatre graphes et remet en question la légitimité du choix du seuil $Q^{\mu_N}_{d,\gamma_1}$ proposé. D'après [Vu 2020], C ne dépend pas de d mais uniquement du problème et $Q^{\mu_N}_{d,\gamma_C}$ doit être fixé pour un d bien choisi, il n'est donc pas surprenant que $C = 1$ ne soit pas un choix optimal et varie autant en fonction du d choisi et du jeu de données.

On remarque également que les valeurs optimales de C sont plus proches pour les mêmes dispositions (aléatoire ou successive) que pour les mêmes jeux de données (deux disques ou deux lunes). Aucune relation évidente ne semble tout du moins ressortir des valeurs optimales obtenues en fonction de d.

Concernant DyCG, les performances obtenues pour les deux jeux de données considérés et leurs deux dispositions sont assez proches de ceux obtenus pour DyCF

avec les valeurs optimales de C, démontrant la force de la méthode sans paramètres.

Cependant, comme le score obtenu avec DyCG, calculé comme une différence de scores $\mathbf{S}_{d,\gamma_1}$ pour deux d distincts, ne partage plus les propriétés de DyCF quant au rapport à la pdf, de plus amples études sont nécessaires pour permettre l'utilisation de DyCG au sein de WOLF avec les définitions de distance et de densité locale.

6.3 Conclusion

Les résultats présentés dans ce chapitre mettent en avant les limites du paramétrage de DyCF dans un cadre général. En s'appuyant sur les propriétés de la fonction de Christoffel, une méthode alternative et s'émancipant du choix de d et C, DyCG, a également été proposée. Celle-ci propose des performances satisfaisantes, mais n'est pas applicable en l'état dans le cadre opérationnel proposé dans le Chapitre 3.

Il serait intéressant d'étudier l'utilisation de DyCG pour différentes définitions de l'anomalie à intégrer à WOLF.

Chapitre 7

Conclusion générale

Ce dernier chapitre conclut le manuscrit. Il rappelle dans un premier temps les objectifs de cette thèse puis synthétise les contributions apportées avant d'ouvrir sur les perspectives.

Sommaire

7.1 Rappel de l'objectif et des verrous

Comme décrit dans le Chapitre 1, qui introduit les différentes notions utilisées dans cette thèse, le besoin traité est l'étude de la fiabilité des données issues de réseaux de capteurs (WSNs) dans des contextes industriels variés. Les spécificités des WSNs et l'absence de connaissances a priori sur les contextes d'application possibles mènent à l'objectif suivant :

Objectif 1 *Proposer une solution, automatisable et agnostique au contexte industriel, à la problématique de détection d'anomalies dans les réseaux de capteurs.*

La synthèse de l'état de l'art réalisé dans le Chapitre 2 permet d'identifier différents verrous scientifiques à rattacher à cet objectif industriel. On y présente différentes méthodes pour la détection d'anomalies dans les WSNs et dans les flux de données selon une taxonomie proposée à cet effet. Cette taxonomie permet en particulier de dresser un tableau récapitulatif décrivant chaque méthode rencontrée.

Un premier verrou concerne la grande variété de méthodes pour traiter le problème de détection d'anomalies dans les WSNs et dans les flux de données :

Verrou 1 *La sélection de méthodes de détection d'anomalies adaptées aux données traitées parmi le large spectre de méthodes non supervisées de l'état de l'art.*

En effet, les méthodes ne sont pas associées à un cas d'application spécifique, en particulier celles pour les flux de données qui ont un usage plus général que celui des WSNs. De plus, l'absence de données labelisées rend l'évaluation des méthodes difficile, ce qui complique la sélection d'une méthode adaptée.

Le problème est le même pour le choix des paramètres, et il faut donc restreindre ces derniers ou, idéalement, s'en émanciper, ce qui mène au second verrou :

Verrou 2 *La suppression, ou au moins la simplification, de la phase de paramétrage de la solution, et donc des méthodes qui la composent.*

L'état de l'art montre aussi qu'une solution adaptée à l'objectif formulé doit respecter certaines contraintes, décrites par le verrou suivant :

Verrou 3 *Une méthode applicable aux réseaux de capteurs doit être composable, satisfaire les contraintes des flux de données et, a minima, tenir compte des dépendances d'attributs à travers une étude multivariée et des dépendances temporelles par un apprentissage en ligne sur des plages temporelles suffisamment larges.*

Le caractère compositionnel doit permettre de tenir compte des dépendances spatiales, avec des modèles locaux et globaux, mais aussi de réduire la charge en communications. Les trois formes de dépendances, temporelles, spatiales et d'attributs, sont importantes, notamment pour différencier les deux types d'anomalies des WSNs présentés dans le Chapitre 1, à savoir les erreurs et les évènements d'intérêt, dans une phase de diagnostic en aval de la détection.

7.2 Synthèse des contributions

Cette section synthétise les contributions réalisées dans ce manuscrit et les discussions déjà présentées dans les chapitres associés.

Le Chapitre 3 propose un cadre opérationnel, appelé WOLF, pour la détection d'anomalies s'accompagnant de WOLF-Eval, une approche pour l'évaluation non supervisée. En observant que certains articles présentaient des méthodes rattachées à une définition particulière de l'anomalie, en particulier dans le cas d'anomalies de distance, WOLF a été décrit comme une boîte à outils où chaque méthode est rattachée à la définition des anomalies qu'elle détecte. WOLF permet également de régler en partie les Verrous 1 et 2 à travers les points suivants :

- les méthodes doivent être rattachées à une définition spécifique, et sous certaines contraintes, leur choix est réduit,
- grâce à WOLF-Eval, les méthodes (Verrou 1) et leurs paramètres (Verrou 2) peuvent être comparées malgré le caractère non supervisé de l'application, il est donc possible d'automatiser leur sélection, à l'exception du seuil sur le score qui n'est pas pris en compte pour l'évaluation mais seulement pour affecter l'étiquette d'anomalie aux échantillons.

En outre, dans le Chapitre 3, une implémentation de WOLF pour les WSNs est présentée, s'appuyant sur la KDE, sous le nom de WOLF-KDE. Le choix de la KDE comme base pour la détection d'anomalies permet de tenir compte du Verrou 3 concernant les contraintes des WSNs car :

- les modèles de KDE sont composables, c'est-à-dire qu'on peut créer un modèle global à partir de deux modèles locaux ; il suffit alors de concaténer la liste des centres de noyaux et de combiner les variances,

— les trois méthodes utilisées, basées sur la KDE, sont à la fois multivariées et peuvent tenir compte des dépendances temporelles grâce à l'apprentissage en ligne sur des fenêtre glissantes.

Le cadre opérationnel WOLF ne repose que sur trois définitions pour le moment, mais il serait intéressant d'en intégrer de nouvelles. De plus, si les définitions de distance et de densité locale ont l'avantage de pouvoir être évaluées avec WOLF-Eval sur n'importe quel jeu de données, ce n'est pas le cas de la définition statistique qui nécessite de connaître la densité de probabilité (pdf) associée au jeu de données. Ainsi, le Verrou 2 reste à traiter puisqu'on ne peut pas fixer automatiquement les paramètres sans méthodes d'évaluation.

Pour maintenir une précision satisfaisante de la détection d'anomalies malgré l'automatisation du choix des méthodes et des paramètres, plusieurs définitions sont considérées, avec pour chacune une méthode associée. Cependant, une approche est nécessaire pour prendre la décision à partir des scores d'anomalies des trois méthodes.

Le Chapitre 4 décrit une adaptation de la méthode SuMeRI, qui permet théoriquement de faciliter le paramétrage des méthodes dans un cadre d'application hors ligne, pour WOLF. Cette adaptation, déclinée en SuMeLInk et SuMeLInd, perd malheureusement la faculté d'aide au paramétrage de SuMeRI. En revanche, elle offre une approche pour l'intégration des différentes méthodes de WOLF.

Cependant, l'approche pour déterminer la décision à partir des différentes méthodes est pour le moment arbitraire, et l'apport de SuMeLInk par rapport à SuMeLInd est peu convaincant, notamment à cause de l'introduction d'un délai dans l'apprentissage de nouveaux comportements.

Le Chapitre 5 propose une alternative aux méthodes basées sur un modèle de KDE dans WOLF-KDE avec l'utilisation de la fonction de Christoffel empirique (FCE) et son implémentation dans WOLF sous le nom de WOLF-DyCF. WOLF-DyCF permet de traiter le Verrou 2 en limitant théoriquement le paramétrage par rapport à WOLF-KDE, tout en pouvant être évaluée de manière équivalente avec WOLF-Eval. Le Verrou 3 est également traité puisque la matrice des moments, sur laquelle repose la FCE, peut être combinée et permet donc d'obtenir un modèle global à partir de modèles locaux. De plus, l'analyse est multivariée et toutes les instances peuvent être considérées, prenant donc en compte l'aspect temporel. On peut également choisir, si jugé nécessaire, d'intégrer un facteur d'oubli pour accorder moins de poids aux instances les plus anciennes.

Enfin, dans le Chapitre 6, le Verrou 2 est plus amplement traité avec la proposition de DyCG, une méthode sans paramètres.

Les méthodes basées sur la FCE proposées, DyCF et DyCG, souffrent tout de même de certaines limites :

— le seuil sur le score de DyCF peut théoriquement être automatisé, mais les résultats obtenus ne sont pas optimaux,
— la complexité explose avec le degré d et le nombre de variables p,
— DyCG lève les précédentes limitations et surpasse DyCF par les résultats obtenus tout en étant une méthode sans aucun paramétrage. Cependant,

elle ne peut pas être intégrée à WOLF en l'état.

7.3 Perspectives

Comme indiqué précédemment, WOLF est pour l'instant limité en termes de définitions de l'anomalie considérées. Une première perspective d'amélioration serait d'intégrer de nouvelles définitions, ce qui pourrait également permettre d'intégrer d'autres méthodes. Néanmoins, les nouvelles définitions doivent respecter les conditions fixées dans le Chapitre 3 et, idéalement, permettre une évaluation au sein de WOLF-Eval à partir des données réelles du cas d'application. En effet, bien que ce soit le cas pour les définitions de distance et de densité locale, ça ne l'est pas pour les anomalies statistiques pour lesquelles il est nécessaire de connaître également la fonction de densité de probabilité. Dans ce cas, une solution peut être d'évaluer les méthodes avec un jeu de données synthétique proche du jeu de données réel, et dont on connaît la pdf associée.

Les adaptations de SuMeRI décrites proposent une première solution de décision sur l'anormalité d'une instance en fonction des scores établis par les différentes méthodes de WOLF. Cependant, une perspective d'évolution consisterait à travailler sur cette prise de décision et, plus largement, sur toutes les étapes menant à un diagnostic facilité par un opérateur en aval de la détection d'anomalies.

En ce qui concerne les méthodes basées sur la FCE, des travaux complémentaires sont nécessaires pour :

— régler le problème d'instabilités observées en cas de données décentrées,
— relier le nombre d'instances dans un voisinage à l'intégrale sous la FCE, en passant possiblement par la FC modifiée de [Lasserre 2023],
— permettre l'utilisation de DyCG au sein de WOLF.

En dehors des perspectives concernant les travaux présentés, certains éléments discutés mais non traités mériteraient d'être étudiés. En particulier, on a supposé que l'étude des dépendances permettrait de différencier les erreurs des évènements d'intérêt, mais cette hypothèse doit être vérifiée avec des tests plus complets. Cependant, il faudrait pour cela posséder des données labellisées avec les deux types d'anomalies. Ce premier point en soulève un second concernant les différentes dépendances considérées. Aussi, il serait intéressant d'intégrer des méthodes étudiant ces dépendances plus spécifiquement.

Aussi, bien que les anomalies contextuelles et collectives, définies dans la Section 1.2.2.1, puissent être ramenées à des anomalies ponctuelles en travaillant sur les attributs du jeu de données, il faut souvent grandement augmenter le nombre d'attributs, ce qui ne garantit plus l'hypothèse selon laquelle on aura toujours un faible nombre d'attributs. Les méthodes proposées pourraient alors ne plus être viables, sans compter que le nombre de modèles en mémoire se multiplierait.

Pour conclure, les travaux présentés ici représentent une première approche, originale et ouvrant de multiples perspectives, du problème de la détection d'anomalies non supervisée, automatisable et agnostique, dans les réseaux de capteurs.

Bibliographie

[Abadi 2003] D. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, C. Erwin, E. Galvez, M. Hatoun, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing, R. Yan et S. Zdonik. *Aurora : a data stream management system.* Dans Proceedings of the 2003 ACM SIGMOD international conference on Management of data, SIGMOD '03, page 666, New York, NY, USA, juin 2003. Association for Computing Machinery. (Cité en page 15.)

[Abid 2017] Aymen Abid, Abdennaceur Kachouri et Adel Mahfoudhi. *Outlier detection for wireless sensor networks using density-based clustering approach.* IET Wireless Sensor Systems, vol. 7, no. 4, pages 83–90, 2017. (Cité en pages 48 et 64.)

[Aggarwal 2003] Charu C. Aggarwal, Philip S. Yu, Jiawei Han et Jianyong Wang. - *A Framework for Clustering Evolving Data Streams.* Dans Johann-Christoph Freytag, Peter Lockemann, Serge Abiteboul, Michael Carey, Patricia Selinger et Andreas Heuer, editeurs, Proceedings 2003 VLDB Conference, pages 81–92. Morgan Kaufmann, San Francisco, janvier 2003. (Cité en page 46.)

[Ahmadi Livani 2013] Mohammad Ahmadi Livani, Mahdi Abadi, Meysam Alikhany et Meisam Yadollahzadeh Tabari. *Outlier detection in wireless sensor networks using distributed principal component analysis.* Journal of AI and Data Mining, vol. 1, no. 1, pages 1–11, 2013. (Cité en pages 56, 57 et 64.)

[Akyildiz 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci. *Wireless sensor networks : a survey.* Computer Networks, vol. 38, no. 4, pages 393–422, mars 2002. (Cité en page 3.)

[Angiulli 2005] F. Angiulli et C. Pizzuti. *Outlier mining in large high-dimensional data sets.* IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 2, pages 203–215, février 2005. (Cité en pages 40 et 78.)

[Angiulli 2007] Fabrizio Angiulli et Fabio Fassetti. *Detecting distance-based outliers in streams of data.* Dans Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM '07, pages 811–820, New York, NY, USA, novembre 2007. Association for Computing Machinery. (Cité en page 40.)

[Auslander 2011] Bryan Auslander, Kalyan Moy Gupta et David W. Aha. *A comparative evaluation of anomaly detection algorithms for maritime video surveillance.* Dans Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X, volume 8019, page 801907. International Society for Optics and Photonics, juin 2011. (Cité en page 7.)

[Ayadi 2017] Aya Ayadi, Oussama Ghorbel, Abdulfattah M. Obeid et Mohamed Abid. *Outlier detection approaches for wireless sensor networks : A sur-*

vey. Computer Networks, vol. 129, pages 319–333, décembre 2017. (Cité en page 30.)

[Barber 2012] David Barber. Bayesian Reasoning and Machine Learning. Cambridge University Press, février 2012. (Cité en page 37.)

[Barnett 1994] Vic Barnett, Toby Lewis et others. Outliers in statistical data, volume 3. Wiley New York, 1994. (Cité en page 6.)

[Bettencourt 2007] Luís M. A. Bettencourt, Aric A. Hagberg et Levi B. Larkey. *Separating the Wheat from the Chaff : Practical Anomaly Detection Schemes in Ecological Applications of Distributed Sensor Networks.* Dans James Aspnes, Christian Scheideler, Anish Arora et Samuel Madden, editeurs, Distributed Computing in Sensor Systems, Lecture Notes in Computer Science, pages 223–239, Berlin, Heidelberg, 2007. Springer. (Cité en pages 36, 37, 40 et 63.)

[Bhat 2020] Srinidhi Bhat et Sanjay Singh. *One-Class Support Vector Machine for Data Streams.* Dans 2020 IEEE REGION 10 CONFERENCE (TENCON), pages 1130–1135, novembre 2020. (Cité en page 56.)

[Boyd 2013] Kendrick Boyd, Kevin H. Eng et C. David Page. *Area under the Precision-Recall Curve : Point Estimates and Confidence Intervals.* Dans Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen et Filip Železný, editeurs, Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, pages 451–466, Berlin, Heidelberg, 2013. Springer. (Cité en page 22.)

[Boyes 2018] Hugh Boyes, Bil Hallaq, Joe Cunningham et Tim Watson. *The industrial internet of things (IIoT) : An analysis framework.* Computers in Industry, vol. 101, pages 1–12, octobre 2018. (Cité en page 3.)

[Branch 2006] J. Branch, B. Szymanski, C. Giannella, Ran Wolff et H. Kargupta. *In-Network Outlier Detection in Wireless Sensor Networks.* Dans 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), pages 51–51, juillet 2006. (Cité en pages 41, 42, 43, 63, 69 et 78.)

[Branch 2013] Joel W. Branch, Chris Giannella, Boleslaw Szymanski, Ran Wolff et Hillol Kargupta. *In-network outlier detection in wireless sensor networks.* Knowledge and Information Systems, vol. 34, no. 1, pages 23–54, janvier 2013. (Cité en pages 42 et 63.)

[Breunig 2000] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng et Jörg Sander. *LOF : identifying density-based local outliers.* Dans Proceedings of the 2000 ACM SIGMOD international conference on Management of data, SIGMOD '00, pages 93–104, New York, NY, USA, mai 2000. Association for Computing Machinery. (Cité en pages 12, 44 et 84.)

[Campos 2016] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent et Michael E. Houle. *On the evaluation of unsupervised outlier detection : measures, datasets, and an empirical study.* Data Mining and Knowledge Discovery, vol. 30, no. 4, pages 891–927, juillet 2016. (Cité en pages 21 et 23.)

[Cao 2006] Feng Cao, Martin Estert, Weining Qian et Aoying Zhou. *Density-Based Clustering over an Evolving Data Stream with Noise.* Dans Proceedings of the 2006 SIAM International Conference on Data Mining (SDM), Proceedings, pages 328–339. Society for Industrial and Applied Mathematics, avril 2006. (Cité en page 46.)

[Cao 2014] Lei Cao, Di Yang, Qingyang Wang, Yanwei Yu, Jiayuan Wang et Elke A. Rundensteiner. *Scalable distance-based outlier detection over high-volume data streams.* Dans 2014 IEEE 30th International Conference on Data Engineering, pages 76–87, mars 2014. (Cité en page 40.)

[Chandola 2009] Varun Chandola, Arindam Banerjee et Vipin Kumar. *Anomaly detection : A survey.* ACM Computing Surveys, vol. 41, no. 3, pages 15 :1–15 :58, juillet 2009. (Cité en pages 6, 7, 8, 10 et 11.)

[Chatzigiannakis 2006] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou et B. Maglaris. *Hierarchical Anomaly Detection in Distributed Large-Scale Sensor Networks.* Dans 11th IEEE Symposium on Computers and Communications (ISCC'06), pages 761–767, juin 2006. (Cité en pages 56 et 63.)

[Chen 2007] Yixin Chen et Li Tu. *Density-based clustering for real-time stream data.* Dans Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07, pages 133–142, New York, NY, USA, août 2007. Association for Computing Machinery. (Cité en page 46.)

[Ciaccia 2001] Paolo Ciaccia, Marco Patella et Pavel Zezula. *M-tree : An Efficient Access Method for Similarity Search in Metric Spaces.* International conference on very large data bases (VLDB), août 2001. (Cité en pages 80 et 81.)

[Deng 2016] Jeremiah D. Deng. *Online Outlier Detection of Energy Data Streams Using Incremental and Kernel PCA Algorithms.* Dans 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pages 390–397, décembre 2016. (Cité en page 57.)

[Domingues 2018] Rémi Domingues, Maurizio Filippone, Pietro Michiardi et Jihane Zouaoui. *A comparative evaluation of outlier detection algorithms : Experiments and analyses.* Pattern Recognition, vol. 74, pages 406–421, février 2018. (Cité en page 20.)

[Ducharlet 2020] Kévin Ducharlet, Louise Travé-Massuyès, Marie-Véronique Le Lann et Youssef Miloudi. *A Multi-phase Iterative Approach for Anomaly Detection and Its Agnostic Evaluation.* Dans Hamido Fujita, Philippe Fournier-Viger, Moonis Ali et Jun Sasaki, editeurs, Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices, Lecture Notes in Computer Science, pages 505–517, Cham, 2020. Springer International Publishing. (Cité en pages 24, 28, 91, 92, 93 et 109.)

[Ducharlet 2022] Kévin Ducharlet, Louise Travé-Massuyès, Marie-Véronique Le Lann et Youssef Miloudi. *Etude des méthodes de détection d'anomalies non*

supervisées appliquées aux flux de données. juin 2022. (Cité en pages 15, 28 et 29.)

[Dunkl 2001] Charles F. Dunkl et Yuan Xu. Orthogonal Polynomials of Several Variables. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2001. (Cité en page 112.)

[Duraj 2021] Agnieszka Duraj et Piotr S. Szczepaniak. *Outlier Detection in Data Streams — A Comparative Study of Selected Methods*. Procedia Computer Science, vol. 192, pages 2769–2778, janvier 2021. (Cité en page 17.)

[Elnahrawy 2004] Eiman Elnahrawy et Badri Nath. *Context-Aware Sensors*. Dans Holger Karl, Adam Wolisz et Andreas Willig, editeurs, Wireless Sensor Networks, Lecture Notes in Computer Science, pages 77–93, Berlin, Heidelberg, 2004. Springer. (Cité en pages 49, 50, 51, 52, 63 et 69.)

[Ester 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise*. Dans Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, pages 226–231, Portland, Oregon, août 1996. AAAI Press. (Cité en pages 46 et 48.)

[F. Y. Edgeworth 1887] F. Y. Edgeworth. *XLI. On discordant observations*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 23, no. 143, pages 364–375, avril 1887. (Cité en page 6.)

[Feng 2017] Zhen Feng, Jingqi Fu, Dajun Du, Fuqiang Li et Sizhou Sun. *A new approach of anomaly detection in wireless sensor networks using support vector data description*. International Journal of Distributed Sensor Networks, vol. 13, no. 1, page 1550147716686161, janvier 2017. (Cité en page 56.)

[Gao 2011] Jun Gao, Weiming Hu, Zhongfei (Mark) Zhang, Xiaoqin Zhang et Ou Wu. *RKOF : Robust Kernel-Based Local Outlier Detection*. Dans Joshua Zhexue Huang, Longbing Cao et Jaideep Srivastava, editeurs, Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, pages 270–283, Berlin, Heidelberg, 2011. Springer. (Cité en page 84.)

[Goix 2016] Nicolas Goix. *How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms ?* arXiv :1607.01152 [cs, stat], juillet 2016. (Cité en page 23.)

[Goldstein 2012] Markus Goldstein et Andreas Dengel. *Histogram-based Outlier Score (HBOS) : A fast Unsupervised Anomaly Detection Algorithm*. septembre 2012. (Cité en pages 38 et 69.)

[Goldstein 2016] Markus Goldstein et Seiichi Uchida. *A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data*. PLOS ONE, vol. 11, no. 4, page e0152173, avril 2016. (Cité en pages 20 et 21.)

[Hawkins 1974] Douglas M. Hawkins. *The Detection of Errors in Multivariate Data Using Principal Components*. Journal of the American Statistical Association, vol. 69, no. 346, pages 340–344, 1974. (Cité en page 13.)

[Hawkins 1980] D. Hawkins. Identification of Outliers. Monographs on Statistics and Applied Probability. Springer Netherlands, 1980. (Cité en page 6.)

[Hodge 2004] Victoria Hodge et Jim Austin. *A Survey of Outlier Detection Methodologies.* Artificial Intelligence Review, vol. 22, no. 2, pages 85–126, octobre 2004. (Cité en pages 7 et 92.)

[Huang 2020] Jen-Wei Huang, Meng-Xun Zhong et Bijay Prasad Jaysawal. *TADILOF : Time Aware Density-Based Incremental Local Outlier Detection in Data Streams.* Sensors, vol. 20, no. 20, page 5829, janvier 2020. (Cité en pages 45 et 69.)

[Janakiram 2006] D. Janakiram, A.V.U.P. Kumar et Adi Mallikarjuna Reddy V. *Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks.* Dans 2006 1st International Conference on Communication Systems Software & Middleware, pages 1–6, janvier 2006. (Cité en pages 52, 63 et 69.)

[Jemal 2013] Ahmed Jemal et Riadh Ben Halima. *A QoS-Driven Self-Adaptive Architecture For Wireless Sensor Networks.* Dans IEEE International Conference on Enabling Technologies : Infrastructures for Collaborative Enterprises (WETICE), page 6p., Hammamet, Tunisia, juin 2013. (Cité en page 3.)

[Jin 2006] Wen Jin, Anthony K. H. Tung, Jiawei Han et Wei Wang. *Ranking Outliers Using Symmetric Neighborhood Relationship.* Dans Wee-Keong Ng, Masaru Kitsuregawa, Jianzhong Li et Kuiyu Chang, editeurs, Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, pages 577–593, Berlin, Heidelberg, 2006. Springer. (Cité en page 12.)

[Jolliffe 2002] I. T. Jolliffe. Principal Component Analysis. Springer Series in Statistics. Springer-Verlag, New York, 2 édition, 2002. (Cité en page 56.)

[Karimian 2012] Seyed Hesamodin Karimian, Manouchehr Kelarestaghi et Sattar Hashemi. *I-IncLOF : Improved incremental local outlier detection for data streams.* Dans The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), pages 023–028, mai 2012. (Cité en pages 44 et 69.)

[Kingma 2014] Diederik P Kingma et Max Welling. *Stochastic gradient VB and the variational auto-encoder.* Dans Second international conference on learning representations, ICLR, volume 19, page 121, 2014. (Cité en page 11.)

[Knorr 1998] Edwin M. Knorr et Raymond T. Ng. *Algorithms for Mining Distance-Based Outliers in Large Datasets.* Dans Proceedings of the 24rd International Conference on Very Large Data Bases, VLDB '98, pages 392–403, San Francisco, CA, USA, août 1998. Morgan Kaufmann Publishers Inc. (Cité en pages 12, 39, 40, 67, 68 et 78.)

[Kontaki 2011] Maria Kontaki, Anastasios Gounaris, Apostolos N. Papadopoulos, Kostas Tsichlas et Yannis Manolopoulos. *Continuous monitoring of distance-based outliers over data streams.* Dans 2011 IEEE 27th International Confe-

rence on Data Engineering, pages 135–146, avril 2011. (Cité en pages 40, 69, 78 et 80.)

[Krawczyk 2015] Bartosz Krawczyk et Michał Woźniak. *One-class classifiers with incremental learning and forgetting for data streams with concept drift*. Soft Computing, vol. 19, no. 12, pages 3387–3400, décembre 2015. (Cité en page 56.)

[Kristan 2011] Matej Kristan, Aleš Leonardis et Danijel Skočaj. *Multivariate online kernel density estimation with Gaussian kernels*. Pattern Recognition, vol. 44, no. 10, pages 2630–2642, octobre 2011. (Cité en pages 39, 69 et 70.)

[Kumar 2010] Ravi Kumar et Sergei Vassilvitskii. *Generalized distances between rankings*. Dans Proceedings of the 19th international conference on World wide web, WWW '10, pages 571–580, New York, NY, USA, avril 2010. Association for Computing Machinery. (Cité en pages 70 et 71.)

[Langrené 2019] Nicolas Langrené et Xavier Warin. *Fast and Stable Multivariate Kernel Density Estimation by Fast Sum Updating*. Journal of Computational and Graphical Statistics, vol. 28, no. 3, pages 596–608, juillet 2019. (Cité en page 39.)

[Laskov 2004] P. Laskov, C. Schäfer, I. Kotenko et K.-R. Müller. *Intrusion Detection in Unlabeled Data with Quarter-sphere Support Vector Machines :*. vol. 27, no. 4, pages 228–236, décembre 2004. (Cité en page 53.)

[Lasserre 2019] Jean-Bernard Lasserre et Edouard Pauwels. *The empirical Christoffel function with applications in data analysis*. Advances in Computational Mathematics, vol. 45, no. 3, pages 1439–1468, juin 2019. (Cité en pages 112 et 113.)

[Lasserre 2022] Jean Bernard Lasserre, Edouard Pauwels et Mihai Putinar. The Christoffel–Darboux Kernel for Data Analysis. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, 2022. (Cité en pages 112, 113, 114 et 142.)

[Lasserre 2023] Jean-Bernard Lasserre. *A modified Christoffel function and its asymptotic properties*, janvier 2023. (Cité en pages 138 et 152.)

[Lavin 2015] Alexander Lavin et Subutai Ahmad. *Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark*. Dans 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pages 38–44, décembre 2015. (Cité en pages 23 et 66.)

[Laxhammar 2009] Rikard Laxhammar, Goran Falkman et Egils Sviestins. *Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator*. Dans 2009 12th International Conference on Information Fusion, pages 756–763, juillet 2009. (Cité en page 89.)

[Li 2000] Weihua Li, H. Henry Yue, Sergio Valle-Cervantes et S. Joe Qin. *Recursive PCA for adaptive process monitoring*. Journal of Process Control, vol. 10, no. 5, pages 471–486, octobre 2000. (Cité en page 57.)

[Liu 2016] Zongyi Liu, Daniela Dragomirescu, Georges Da Costa et Thierry Monteil. *Dynamic multi-channel allocation mechanism for wireless multimedia sensor networks*. Dans Wireless Days (WD 2016), pages pp. 1–6, Toulouse, France, mars 2016. (Cité en page 4.)

[Marques 2015] Henrique O. Marques, Ricardo J. G. B. Campello, Arthur Zimek et Jörg Sander. *On the internal evaluation of unsupervised outlier detection*. Dans Proceedings of the 27th International Conference on Scientific and Statistical Database Management, SSDBM '15, pages 1–12, New York, NY, USA, juin 2015. Association for Computing Machinery. (Cité en page 23.)

[Martin 2007] R. A. Martin, M. Schwabacher, N. Oza et A. Srivastava. *Comparison of Unsupervised Anomaly Detection Methods for Systems Health Management Using Space Shuttle*. Dans Main Engine Data," Proceedings of the Joint Army Navy NASA Air Force Conference on Propulsion, 2007, 2007. (Cité en page 7.)

[Mathur 2006] Gaurav Mathur, P. Desnoyers, Deepak Ganesan et Prashant Shenoy. *Ultra-low power data storage for sensor networks*. Dans 2006 5th International Conference on Information Processing in Sensor Networks, pages 374–381, avril 2006. (Cité en page 5.)

[Mokrenko 2014] Olesia Mokrenko, Suzanne Lesecq, Warody Lombardi, Diego Puschini, Carolina Albea-Sanchez et Olivier Debicki. *Dynamic Power Management in a Wireless Sensor Network using Predictive Control*. Dans 40th Annual Conference of the IEEE Industrial Electronics Society, 40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, United States, octobre 2014. (Cité en page 3.)

[Moshtaghi 2009] Masud Moshtaghi, Sutharshan Rajasegarar, Christopher Leckie et Shanika Karunasekera. *Anomaly detection by clustering ellipsoids in wireless sensor networks*. Dans 2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pages 331–336, décembre 2009. (Cité en pages 47, 55 et 64.)

[Motwani 2003] Rajeev Motwani, Jennifer Widom, Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Gurmeet Singh Manku, Chris Olston, Justin Rosenstein et Rohit Varma. *Query Processing, Approximation, and Resource Management in a Data Stream Management System*. Dans First Biennial Conference on Innovative Data Systems Research, CIDR 2003, Asilomar, CA, USA, January 5-8, 2003, Online Proceedings. www.cidrdb.org, 2003. (Cité en page 15.)

[Na 2018] Gyoung S. Na, Donghyun Kim et Hwanjo Yu. *DILOF : Effective and Memory Efficient Local Outlier Detection in Data Streams*. Dans Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, pages 1993–2002, New York, NY, USA, juillet 2018. Association for Computing Machinery. (Cité en pages 45 et 69.)

[Neal 1998] Radford M. Neal et Geoffrey E. Hinton. *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*. Dans Michael I. Jordan, editeur, Learning in Graphical Models, NATO ASI Series, pages 355–368. Springer Netherlands, Dordrecht, 1998. (Cité en page 73.)

[Nevai 1986] Paul Nevai. *Géza Freud, orthogonal polynomials and Christoffel functions. A case study*. Journal of Approximation Theory, vol. 48, no. 1, pages 3–167, septembre 1986. (Cité en page 112.)

[O'Reilly 2012] Colin O'Reilly, Alex Gluhak, Muhammad Imran et Sutharshan Rajasegarar. *Online anomaly rate parameter tracking for anomaly detection in wireless sensor networks*. Dans 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pages 191–199, juin 2012. (Cité en page 54.)

[Palpanas 2003] Themistoklis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki et Dimitrios Gunopulos. *Distributed deviation detection in sensor networks*. ACM SIGMOD Record, vol. 32, no. 4, pages 77–82, décembre 2003. (Cité en pages 39, 41, 43, 60, 63, 68, 69, 70 et 78.)

[Papadimitriou 2003] S. Papadimitriou, H. Kitagawa, P. B. Gibbons et C. Faloutsos. *LOCI : fast outlier detection using the local correlation integral*. Dans Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405), pages 315–326, mars 2003. (Cité en pages 12, 40, 45, 84, 85 et 86.)

[Parzen 1962] E. Parzen. *On Estimation of a Probability Density Function and Mode*. 1962. (Cité en page 38.)

[Pokrajac 2007] Dragoljub Pokrajac, Aleksandar Lazarevic et Longin Jan Latecki. *Incremental Local Outlier Detection for Data Streams*. Dans 2007 IEEE Symposium on Computational Intelligence and Data Mining, pages 504–515, mars 2007. (Cité en pages 44 et 69.)

[Rajasegarar 2006] Sutharshan Rajasegarar, Christopher Leckie, Marimuthu Palaniswami et James C. Bezdek. *Distributed Anomaly Detection in Wireless Sensor Networks*. Dans 2006 10th IEEE Singapore International Conference on Communication Systems, pages 1–5, octobre 2006. (Cité en pages 47 et 63.)

[Rajasegarar 2007] S. Rajasegarar, C. Leckie, M. Palaniswami et J. C. Bezdek. *Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks*. Dans 2007 IEEE International Conference on Communications, pages 3864–3869, juin 2007. (Cité en pages 54, 55 et 64.)

[Rajasegarar 2010] Sutharshan Rajasegarar, Christopher Leckie, James C. Bezdek et Marimuthu Palaniswami. *Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Networks*. IEEE Transactions on Information Forensics and Security, vol. 5, no. 3, pages 518–533, septembre 2010. (Cité en pages 55 et 64.)

[Rajasegarar 2014] Sutharshan Rajasegarar, Christopher Leckie et Marimuthu Palaniswami. *Hyperspherical cluster based distributed anomaly detection in wireless sensor networks.* Journal of Parallel and Distributed Computing, vol. 74, no. 1, pages 1833–1847, janvier 2014. (Cité en pages 47 et 63.)

[Ramaswamy 2000] Sridhar Ramaswamy, Rajeev Rastogi et Kyuseok Shim. *Efficient algorithms for mining outliers from large data sets.* ACM SIGMOD Record, vol. 29, no. 2, pages 427–438, mai 2000. (Cité en pages 40 et 78.)

[Ren 2009] Jiadong Ren et Ruiqing Ma. *Density-Based Data Streams Clustering over Sliding Windows.* Dans 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, volume 5, pages 248–252, août 2009. (Cité en page 46.)

[Rezende 2015] Danilo Rezende et Shakir Mohamed. *Variational Inference with Normalizing Flows.* Dans Francis Bach et David Blei, editeurs, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, juillet 2015. PMLR. (Cité en page 11.)

[Ristic 2008] B. Ristic, B. La Scala, M. Morelande et N. Gordon. *Statistical analysis of motion patterns in AIS Data : Anomaly detection and motion prediction.* Dans 2008 11th International Conference on Information Fusion, pages 1–7, juin 2008. (Cité en page 89.)

[Roa 2019] Nathalie Barbosa Roa, Louise Travé-Massuyès et Victor Hugo Grisales. *DyClee : Dynamic clustering for tracking evolving environments.* Pattern Recognition, vol. 94, page 162, octobre 2019. (Cité en page 46.)

[Rose 2015] Karen Rose, Scott Eldridge et Lyman Chapin. *The internet of things : An overview.* The internet society (ISOC), vol. 80, pages 1–50, 2015. (Cité en page 2.)

[Ruff 2018] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller et Marius Kloft. *Deep One-Class Classification.* Dans Jennifer Dy et Andreas Krause, editeurs, Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, juillet 2018. (Cité en page 13.)

[Ruff 2021] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich et Klaus-Robert Müller. *A Unifying Review of Deep and Shallow Anomaly Detection.* Proceedings of the IEEE, vol. 109, no. 5, pages 756–795, mai 2021. (Cité en pages 7, 8, 10, 11, 21 et 22.)

[Sadik 2014] Shiblee Sadik et Le Gruenwald. *Research issues in outlier detection for data streams.* ACM SIGKDD Explorations Newsletter, vol. 15, no. 1, pages 33–40, mars 2014. (Cité en page 16.)

[Sakurada 2014] Mayu Sakurada et Takehisa Yairi. *Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction.* Dans Proceedings of

the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, MLSDA'14, pages 4–11, New York, NY, USA, décembre 2014. Association for Computing Machinery. (Cité en page 13.)

[Salehi 2016] Mahsa Salehi, Christopher Leckie, James C. Bezdek, Tharshan Vaithianathan et Xuyun Zhang. *Fast Memory Efficient Local Outlier Detection in Data Streams*. IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 12, pages 3246–3260, décembre 2016. (Cité en pages 45 et 69.)

[Salehi 2018] Mahsa Salehi et Lida Rashidi. *A Survey on Anomaly detection in Evolving Data : [with Application to Forest Fire Risk Prediction]*. ACM SIGKDD Explorations Newsletter, vol. 20, no. 1, pages 13–23, mai 2018. (Cité en pages 18 et 46.)

[Schubert 2012] Erich Schubert, Remigius Wojdanowski, Arthur Zimek et Hans-Peter Kriegel. *On Evaluation of Outlier Rankings and Outlier Scores*. Dans Proceedings of the 2012 SIAM International Conference on Data Mining (SDM), Proceedings, pages 1047–1058. Society for Industrial and Applied Mathematics, avril 2012. (Cité en pages 22 et 23.)

[Schubert 2014] Erich Schubert, Arthur Zimek et Hans-Peter Kriegel. *Generalized Outlier Detection with Flexible Kernel Density Estimates*. Dans Proceedings of the 2014 SIAM International Conference on Data Mining (SDM), Proceedings, pages 542–550. Society for Industrial and Applied Mathematics, avril 2014. (Cité en pages 84 et 89.)

[Schölkopf 2001] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola et Robert C. Williamson. *Estimating the Support of a High-Dimensional Distribution*. Neural Computation, vol. 13, no. 7, pages 1443–1471, juillet 2001. (Cité en page 13.)

[Scott 1992] David W. Scott. Multivariate Density Estimation : Theory, Practice, and Visualization. Wiley Series in Probability and Statistics. Wiley, 1992. (Cité en pages 39 et 72.)

[Shahid 2012a] Nauman Shahid, Ijaz Haider Naqvi et Saad Bin Qaisar. *Quarter-Sphere SVM : Attribute and Spatio-Temporal correlations based Outlier & Event Detection in wireless sensor networks*. Dans 2012 IEEE Wireless Communications and Networking Conference (WCNC), pages 2048–2053, avril 2012. (Cité en pages 54, 55 et 64.)

[Shahid 2012b] Nauman Shahid, Ijaz Haider Naqvi et Saad Bin Qaisar. *Real time energy efficient approach to Outlier & event detection in wireless sensor networks*. Dans 2012 IEEE International Conference on Communication Systems (ICCS), pages 162–166, novembre 2012. (Cité en pages 54, 55 et 64.)

[Sheng 2007] Bo Sheng, Qun Li, Weizhen Mao et Wen Jin. *Outlier detection in sensor networks*. Dans Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '07, pages 219–228, New York, NY, USA, septembre 2007. Association for Computing Machinery. (Cité en pages 41, 42, 63, 68, 69 et 78.)

[Spearman 1987] C. Spearman. *The Proof and Measurement of Association between Two Things*. The American Journal of Psychology, vol. 100, no. 3/4, pages 441–471, 1987. (Cité en page 70.)

[Subramaniam 2006] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki et D. Gunopulos. *Online outlier detection in sensor data using non-parametric models*. Dans Proceedings of the 32nd international conference on Very large data bases, VLDB '06, pages 187–198, Seoul, Korea, septembre 2006. VLDB Endowment. (Cité en pages 39, 41, 43, 45, 60, 63, 68, 69, 70, 78, 84 et 86.)

[Tang 2002] Jian Tang, Zhixiang Chen, Ada Wai-chee Fu et David W. Cheung. *Enhancing Effectiveness of Outlier Detections for Low Density Patterns*. Dans Ming-Syan Chen, Philip S. Yu et Bing Liu, editeurs, Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, pages 535–548, Berlin, Heidelberg, 2002. Springer. (Cité en page 12.)

[Tang 2017] Bo Tang et Haibo He. *A local density-based approach for outlier detection*. Neurocomputing, vol. 241, pages 171–180, juin 2017. (Cité en page 84.)

[Tax 2004] David M.J. Tax et Robert P.W. Duin. *Support Vector Data Description*. Machine Learning, vol. 54, no. 1, pages 45–66, janvier 2004. (Cité en page 53.)

[Thakkar 2016] Pooja Thakkar, Jay Vala et Vishal Prajapati. *Survey on outlier detection in data stream*. Int. J. Comput. Appl, vol. 136, pages 13–16, 2016. (Cité en pages 37 et 46.)

[Titouna 2015] Chafiq Titouna, Makhlouf Aliouat et Mourad Gueroui. *Outlier Detection Approach Using Bayes Classifiers in Wireless Sensor Networks*. Wireless Personal Communications, vol. 85, no. 3, pages 1009–1023, décembre 2015. (Cité en pages 49, 50, 51 et 64.)

[Titouna 2019] Chafiq Titouna, Farid Naït-Abdesselam et Ashfaq Khokhar. *DODS : A Distributed Outlier Detection Scheme for Wireless Sensor Networks*. Computer Networks, vol. 161, pages 93–101, octobre 2019. (Cité en pages 49, 50, 51 et 64.)

[Tran 2016] Luan Tran, Liyue Fan et Cyrus Shahabi. *Distance-based outlier detection in data streams*. Proceedings of the VLDB Endowment, vol. 9, no. 12, pages 1089–1100, août 2016. (Cité en page 40.)

[Vu 2020] Mai Trang Vu, François Bachoc et Edouard Pauwels. *Rate of convergence for geometric inference based on the empirical Christoffel function*, mai 2020. (Cité en pages 114, 141 et 146.)

[Wang 2019] Hongzhi Wang, Mohamed Jaward Bah et Mohamed Hammad. *Progress in Outlier Detection Techniques : A Survey*. IEEE Access, vol. 7, pages 107964–108000, 2019. (Cité en pages 10, 11 et 13.)

[Wu 2007] Weili Wu, Xiuzhen Cheng, Min Ding, Kai Xing, Fang Liu et Ping Deng. *Localized Outlying and Boundary Data Detection in Sensor Networks*. IEEE

Transactions on Knowledge and Data Engineering, vol. 19, no. 8, pages 1145–1157, août 2007. (Cité en pages 36, 37 et 63.)

[Xie 2011] Miao Xie, Song Han, Biming Tian et Sazia Parvin. *Anomaly detection in wireless sensor networks : A survey.* Journal of Network and Computer Applications, vol. 34, no. 4, pages 1302–1325, juillet 2011. (Cité en pages 30 et 69.)

[Xie 2013] Miao Xie, Jiankun Hu, Song Han et Hsiao-Hwa Chen. *Scalable Hypergrid k-NN-Based Online Anomaly Detection in Wireless Sensor Networks.* IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 8, pages 1661–1670, août 2013. (Cité en pages 41, 43 et 64.)

[Yamanishi 2004] Kenji Yamanishi, Jun-ichi Takeuchi, Graham Williams et Peter Milne. *On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms.* Data Mining and Knowledge Discovery, vol. 8, no. 3, pages 275–300, mai 2004. (Cité en pages 37, 39, 69, 70, 73, 74 et 75.)

[Yang 2009] Di Yang, Elke A. Rundensteiner et Matthew O. Ward. *Neighbor-based pattern detection for windows over streaming data.* Dans Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology, EDBT '09, pages 529–540, New York, NY, USA, mars 2009. Association for Computing Machinery. (Cité en page 40.)

[Yao 2015] Haiqing Yao, Heng Cao et Jin Li. *Comprehensive Outlier Detection in Wireless Sensor Network with Fast Optimization Algorithm of Classification Model.* International Journal of Distributed Sensor Networks, vol. 11, no. 7, page 398761, juillet 2015. (Cité en pages 54 et 64.)

[Zhang 1996] Tian Zhang, Raghu Ramakrishnan et Miron Livny. *BIRCH : an efficient data clustering method for very large databases.* ACM SIGMOD Record, vol. 25, no. 2, pages 103–114, juin 1996. (Cité en page 46.)

[Zhang 2007] Kejia Zhang, Shengfei Shi, Hong Gao et Jianzhong Li. *Unsupervised Outlier Detection in Sensor Networks Using Aggregation Tree.* Dans Reda Alhajj, Hong Gao, Jianzhong Li, Xue Li et Osmar R. Zaïane, editeurs, Advanced Data Mining and Applications, Lecture Notes in Computer Science, pages 158–169, Berlin, Heidelberg, 2007. Springer. (Cité en pages 41, 42, 43, 63, 69 et 78.)

[Zhang 2008] Yang Zhang, Nirvana Meratnia et Paul Havinga. *An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine.* Dans 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pages 151–156, décembre 2008. (Cité en pages 54 et 55.)

[Zhang 2009] Yang Zhang, Nirvana Meratnia et Paul Havinga. *Adaptive and Online One-Class Support Vector Machine-Based Outlier Detection Techniques for Wireless Sensor Networks.* Dans 2009 International Conference on Advanced

Information Networking and Applications Workshops, pages 990–995, mai 2009. (Cité en pages 54, 55 et 64.)

[Zhang 2010] Yang Zhang, Nirvana Meratnia et Paul Havinga. *Outlier Detection Techniques for Wireless Sensor Networks : A Survey.* IEEE Communications Surveys & Tutorials, vol. 12, no. 2, pages 159–170, 2010. (Cité en page 30.)

[Zhang 2013a] Ji Zhang. *Advancements of outlier detection : a survey.* ICST Transactions on Scalable Information Systems, vol. 13, no. 1, pages 1–26, février 2013. (Cité en pages 10, 11 et 46.)

[Zhang 2013b] Yang Zhang, Nirvana Meratnia et Paul J. M. Havinga. *Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine.* Ad Hoc Networks, vol. 11, no. 3, pages 1062–1074, mai 2013. (Cité en pages 55 et 64.)

[Zhao 2003] Feng Zhao, Jie Liu, Juan Liu, L. Guibas et J. Reich. *Collaborative signal and information processing : an information-directed approach.* Proceedings of the IEEE, vol. 91, no. 8, pages 1199–1209, août 2003. (Cité en pages 5 et 58.)

[Zhao 2006] Haitao Zhao, Pong Chi Yuen et J.T. Kwok. *A novel incremental principal component analysis and its application for face recognition.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 36, no. 4, pages 873–886, août 2006. (Cité en page 57.)

[Zimek 2018] Arthur Zimek et Peter Filzmoser. *There and back again : Outlier detection between statistical reasoning and data mining algorithms.* WIREs Data Mining and Knowledge Discovery, vol. 8, no. 6, page e1280, 2018. (Cité en pages 6 et 22.)

Résumé :

Les réseaux de capteurs sans fils ont connu un intérêt grandissant depuis le développement de l'Internet des Objets avec des enjeux économiques majeurs, notamment dans les contextes spécifiques de la gestion des bâtiments et la maintenance des équipements. Ces réseaux sont composés d'une quantité souvent importante de nœuds appelés capteurs qui sont peu coûteux mais qui ont d'importantes contraintes en ressources mémoire, calculatoire et énergétique. Ces ressources limitées réduisent la fiabilité des mesures réalisées par les capteurs, propriété pourtant critique dans la majorité des cas d'applications.

Aussi, de nombreux travaux ont vu le jour pour assurer la qualité des mesures en détectant les défauts au niveau des capteurs, mais détectant également des états particuliers au sein des réseaux de capteurs. En particulier, le domaine de la détection d'anomalies a grandement contribué à la détection de ces deux types d'évènements.

A la suite d'une étude de l'état de l'art des méthodes de détection d'anomalie dans les réseaux de capteurs et les flux de données qu'ils génèrent, nos travaux de recherche visent à proposer une solution générique à travers un cadre opérationnel, appelé WOLF, pour la détection d'anomalie non supervisée dans des flux de données. Nous formulons ainsi plusieurs définitions des anomalies s'appuyant sur l'état de l'art et nous proposons SuMeLI, une approche pour détecter ces différentes anomalies de manière intégrée. En faisant le constat du coût élevé du paramétrage des méthodes au sein d'une solution générique, nous formalisons deux nouvelles méthodes de détection d'anomalie dans un contexte d'IA hybride. Celles-ci exploitent les données en s'appuyant sur des résultats de la théorie de l'approximation et des polynômes orthogonaux. Les deux méthodes, nommées DyCF et DyCG, exploitent les propriétés de la fonction de Christoffel à différents niveaux, obtenant ainsi une efficacité similaire aux meilleures méthodes de l'état de l'art avec un paramétrage réduit pour DyCF et sans paramétrage pour DyCG.

Mots clés : Détection d'anomalies, Analyse de données, Réseaux de capteurs, Internet des objets, Apprentissage automatique, Statistiques

Abstract :

Wireless Sensor Networks (WSNs) have known a growing interest since the raise of Internet of Things with major economic stakes, especially in the application cases of building and assets managements. Those networks are composed of a large number of low-cost sensor nodes which are constrained in memory, computational capabilities and power sources. Those limited resources reduce the reliability of the measurements made by sensors, a critical property in most applications.

Also, many works have been done to ensure measurements quality by detecting errors in sensors as well as events of interest within the WSN or the environment it is sensing. Particularly, the field of outlier detection has greatly contributed to the detection of these two types of events.

Following a study of the state of the art of outlier detection methods in WSNs and the data streams they generate, our research work aims at proposing a generic solution through a framework for unsupervised outlier detection in data streams called WOLF. We formulate several definitions of anomalies based on the state of the art and we propose SuMeLI, a global approach to detect these different anomalies. By noting the high cost of parameterizing methods within a generic solution, we finally formalize two new methods for outlier detection in an AI hybrid context. These methods are data-based and exploit results from approximation theory and orthogonal polynomials. Both methods, named DyCF and DyCG, leverage the properties of the Christoffel Function at different levels, achieving efficiency similar to the state of the art with reduced parameters for DyCF and without any for DyCG.

Keywords : Outlier detection, Data mining, Wireless Sensor Networks, Internet of Things, Machine Learning, Statistics