République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Sétif 1 Ferhat Abbas
Faculté des Sciences
Département d'Informatique

Clustering dans l'IoT basé Machine Learning

Une thèse Présentée par :

Malha Merah

Pour l'obtention du diplôme de Doctorat LMD en Informatique
Option : Sûreté de fonctionnement dans les réseaux dynamiques

Soutenue devant le jury composé de :

Pr. Chafia KARA-MOHAMED	Université Sétif 1 Ferhat Abbas	Présidente
Pr. Zibouda ALIOUAT	Université Sétif 1 Ferhat Abbas	Directrice de Thèse
Pr. Hakim MABED	Université de Bourgogne Franche-Comté	Co-Directeur de Thèse
Dr. Chirihane GHERBI	Université Sétif 1 Ferhat Abbas	Examinatrice
Dr. Abdelmalek BOUDRIES	Université de Béjaïa	Examinateur

publiquement le : 17 Avril 2025

Résumé

L'Internet des objets (IoT) connecte des milliards de dispositifs électroniques hétérogènes, permettant une communication fluide et une collecte massive de données. Toutefois, les ressources limitées en mémoire et en énergie de ces dispositifs posent des défis majeurs, notamment en matière de gestion des données et d'efficacité énergétique. Le clustering, en tant que méthode essentielle pour organiser et structurer les réseaux IoT, joue un rôle central dans l'amélioration de l'efficacité énergétique et de la gestion des ressources. Cette technique consiste à regrouper les dispositifs en clusters et à élire un chef de groupe (CH) pour chaque cluster, afin de réduire les coûts de communication, de prolonger la durée de vie des dispositifs et d'optimiser l'équilibrage de la charge dans les réseaux. À ce jour, de nombreuses approches de clustering ont été proposées pour améliorer les performances de la collecte de données dans l'IoT. Cependant, la plupart d'entre elles se concentrent sur le partitionnement des réseaux avec des topologies statiques, ce qui limite leur efficacité pour gérer les aspects incertains et dynamiques des réseaux IoT. Ce constat motive l'exploration de nouvelles solutions intelligentes pour répondre à ces limitations.

Cette thèse vise à développer des solutions novatrices de clustering exploitant l'apprentissage automatique (ML) pour les réseaux IoT, avec un accent particulier sur les réseaux de capteurs sans fil (WSNs), qui sont au cœur de l'IoT, ainsi que sur l'edge computing, dont les performances de calcul sont le plus souvent mises à profit des applications IoT. La première contribution de ce travail propose l'intégration de l'algorithme U-k-means, une méthode de clustering capable de déterminer automatiquement le nombre optimal de clusters dans les WSNs, où ce paramètre est souvent inconnu dans la plupart des applications. Cette méthode est combinée avec un algorithme génétique (GA) pour une sélection efficace des CHs. La deuxième contribution présente un mécanisme de routage optimisé par le Q-learning, qui sélectionne dynamiquement les CHs dans les WSNs pour équilibrer la charge tout en maximisant la performance du réseau. Enfin, la troisième contribution introduit une méthode d'optimisation avancée pour l'équilibrage

de la charge dans des réseaux IoT supervisés par des serveurs edge. Cette méthode s'appuie sur l'apprentissage profond (DL) pour anticiper les variations du trafic, associé à un GA pour une distribution intelligente de la charge entre les serveurs.

Dans l'ensemble, ce travail illustre comment l'intégration du ML peut transformer la gestion des réseaux IoT en proposant des solutions intelligentes, efficaces et adaptées aux contraintes de ces réseaux.

Mots-clés : Internet des Objets, Réseaux de Capteurs Sans Fil, Edge Computing, Apprentissage Automatique, Clustering, Clustering basé sur l'Apprentissage Automatique, Efficacité Énergétique, Sélection de Chef de Cluster, Équilibrage de Charge.

Abstract

The Internet of Things (IoT) connects billions of heterogeneous electronic devices, enabling seamless communication and massive data collection. However, the limited memory and energy resources of these devices pose significant challenges, particularly in data management and energy efficiency. Clustering, as a key method for organizing and structuring IoT networks, plays a central role in improving energy efficiency and resource management. This technique involves grouping devices into clusters and electing a cluster head (CH) for each cluster to reduce communication costs, extend device lifetimes, and optimize load balancing within the networks. To date, numerous clustering approaches have been proposed to enhance data collection performance in IoT. However, most of these methods focus on partitioning networks with static topologies, limiting their effectiveness in addressing the uncertain and dynamic aspects of IoT networks. This observation motivates the exploration of new intelligent solutions to overcome these limitations.

This thesis aims to develop innovative clustering solutions leveraging machine learning (ML) for IoT networks, with a particular focus on wireless sensor networks (WSNs), which are at the core of IoT, as well as edge computing, whose computational capabilities are often utilized to enhance IoT applications. The first contribution of this work proposes integrating the U-k-means algorithm, a clustering method capable of automatically determining the optimal number of clusters in WSNs, where this parameter is often unknown in most applications. This method is combined with a genetic algorithm (GA) for efficient CH selection. The second contribution introduces a routing mechanism optimized with Q-learning, which dynamically selects CHs in WSNs to balance the load while maximizing network performance. Finally, the third contribution presents an advanced optimization method for load balancing in IoT networks managed by edge servers. This method relies on deep learning to predict traffic variations, coupled with a GA for the intelligent distribution of the load across servers.

Overall, this work demonstrates how integrating ML can transform IoT network

management by providing intelligent, efficient, and constraint-adapted solutions.

Keywords : Internet of Things, Wireless Sensor Networks, Edge Computing, Machine Learning, Clustering, ML-Based Clustering, Energy Efficiency, Cluster Head Selection, Load Balancing.

ملخص

يربط إنترنت الأشياء (IoT) مليارات الأجهزة الإلكترونية غير المتجانسة، مما يتيح تواصلاً سلسًا وجمعًا ضخمًا للبيانات. ومع ذلك، فإن الموارد المحدودة للذاكرة والطاقة في هذه الأجهزة تطرح تحديات كبيرة، خاصة في إدارة البيانات وكفاءة استهلاك الطاقة. يلعب التجميع، كطريقة أساسية لتنظيم وهيكلة شبكات إنترنت الأشياء، دورًا محوريًا في تحسين كفاءة استهلاك الطاقة وإدارة الموارد. تتضمن هذه التقنية تجميع الأجهزة في مجموعات واختيار رئيس لكل مجموعة (CH) لتقليل تكاليف الاتصال، وإطالة عمر الأجهزة، وتحسين توزيع الحمل داخل الشبكات. حتى الآن، تم اقتراح العديد من أساليب التجميع لتحسين أداء جمع البيانات في إنترنت الأشياء. ومع ذلك، تركز معظم هذه الأساليب على تقسيم الشبكات إلى هياكل ثابتة، مما يحد من فعاليتها في التعامل مع الجوانب غير المؤكدة والديناميكية لشبكات إنترنت الأشياء. يدفع هذا الواقع إلى استكشاف حلول ذكية جديدة لمعالجة هذه القيود.

تهدف هذه الأطروحة إلى تطوير حلول مبتكرة للتجميع تعتمد على التعلم الآلي (ML) لشبكات إنترنت الأشياء، مع التركيز بشكل خاص على شبكات الاستشعار اللاسلكية $(WSNs)$ ، التي تُعد جوهر إنترنت الأشياء، بالإضافة إلى الحوسبة الطرفية، التي تُستغل قدراتها الحسابية غالبا لتحسين تطبيقات إنترنت الأشياء. يتمثل الإسهام الأول لهذا العمل في اقتراح دمج خوارزمية $U-k-means$ ، وهي طريقة تجميع قادرة على تحديد العدد الأمثل للمجموعات تلقائيًا في شبكات $WSNs$ ، حيث غالبًا ما يكون هذا المعطى غير معروف في معظم التطبيقات. تُدمج هذه الطريقة مع خوارزمية وراثية (GA) لاختيار رؤساء المجموعات بكفاءة. أما الإسهام الثاني، فيقدم آلية توجيه مُحسّنة باستخدام خوارزمية $Q-learning$ ، حيث يتم اختيار رؤساء للمجموعات ديناميكيًا في شبكات $WSNs$ لتوزيع الحمل بشكل متوازن مع تحقيق أقصى أداء للشبكة. وأخيرًا، يقدم الإسهام الثالث طريقة تحسين متقدمة لتوزيع الأحمال في شبكات إنترنت

ملخص

الأشياء التي تُدار بواسطة خوادم الحوسبة الطرفية. تعتمد هذه الطريقة على التعلم العميق (DL) للتنبؤ بالتغيرات في حركة البيانات، مع دمجها ب GA لتوزيع الأحمال بذكاء بين الخوادم.

بشكل عام، يوضح هذا العمل كيف يمكن أن يُحدث دمج تقنيات ML تحولاً في إدارة شبكات إنترنت الأشياء من خلال توفير حلول ذكية وفعالة ومتوافقة مع قيود هذه الشبكات.

الكلمات المفتاحية إنترنت الأشياء، شبكات الاستشعار اللاسلكية، الحوسبة الطرفية، التعلم الآلي، التجميع، التجميع القائم على التعلم الآلي، كفاءة الطاقة، اختيار قائد المجموعة، توازن الأحمال.

Remerciements & dédicace

Avant tout, je tiens à exprimer ma profonde gratitude à **Allah**, le Tout-Puissant, pour m'avoir accordé la force, la patience et la persévérance nécessaires à la réalisation de cette thèse.

Mes sincères remerciements vont à ma directrice de thèse, **Pr. Zibouda Aliouat**, pour son encadrement exceptionnel, ses conseils avisés et sa patience tout au long de ce parcours. La réalisation de ce travail n'aurait pas été possible sans son soutien précieux et ses directives éclairées.

Je tiens également à adresser mes remerciements à mon co-encadrant de thèse, **Dr. Hakim Mabed**, pour sa vision internationale et son approche attentive. J'ai particulièrement apprécié la clarté de ses directives et la qualité de nos échanges, qui ont enrichi ma réflexion et contribué de manière significative à l'avancement de ce travail.

Je voudrais rendre hommage à ma défunte maman, **Dr. Zinouba Taïri**, qu'Allah lui accorde son paradis. Ma Mounette, celle qui m'a le plus inspiré. Une grande femme, qui n'est plus parmi nous, mais dont l'amour et le souvenir continuent d'éclairer mon chemin. Puisse cette thèse être un hommage à sa mémoire.

Mes plus chaleureux remerciements vont également à ma famille : mon **papa**, ma sœur **Meriem** et mon frère **Nadjib**, pour leur amour, leur soutien indéfectible et leur présence.

Je tiens à remercier mes amis et mes collègues, qui m'ont apporté leur soutien et leurs encouragements dans les moments les plus intenses de cette aventure. Leur présence et leur bienveillance ont rendu ce parcours bien plus agréable. Ils et elles se reconnaîtront. Merci **Amira** d'avoir été là dans les pires comme dans les meilleurs moments. Merci **Imène** pour ton soutien et ton aide inconditionnelle.

Je souhaite également adresser un mot de reconnaissance à l'ensemble du personnel de **Frequency** pour leur présence et leur soutien au quotidien. Un remerciement particulier est adressé aux gérants **Sofiane** et **Fares**.

Je remercie également les membres du jury pour avoir accepté d'évaluer mon travail. Enfin, je tiens à remercier toute personne qui a contribué de près ou de loin à la réalisation de ce travail.

Table des matières

Liste des tableaux

Table des figures

Liste des publications

Revues internationales

1. M. Merah, Z. Aliouat, M. Mabed, “Dynamic Load Balancing of Traffic in the IoT Edge Computing Environment using a Clustering Approach based on Deep Learning and Genetic Algorithms”, Cluster Computing [1].
2. M. MERAH, Z. Aliouat, M. Mabed, “Cross-joint K-means and genetic scheme for internet of things sensor clustering”, International Journal of Sensor Networks [2].
3. M. Merah, Z. Aliouat, Y. Harbi, M. S. Batta, “Machine learning-based clustering protocols for Internet of Things networks : An overview”, International Journal of Communication Systems [3].

Conférences internationales

1. M. Merah, Z. Aliouat, H. Mabed, "A novel cluster head selection algorithm based on Q-learning for Internet of Things networks", 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), Djelfa, Algerie, 2024 [4].
2. M. Merah, Z. Aliouat, "Unsupervised K-means for Energy Conservation in IoT Networks", 2022 First International Conference on Computer Communications and Intelligent Systems (I3CIS), Jijel, Algerie, 2022 [5].
3. M. Merah, Z. Aliouat, M. S. Batta, "A Hybrid Neural Network and Graph theory based Clustering Protocol for dynamic IoT Network", International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algerie, 2022 [6].
4. M. Merah, Z. Aliouat, C. Kara-Mohamed, "An energy efficient self organizing map based clustering protocol for iot networks", 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecom-

munications (SETIT), Sfax, Tunisie, 2022 [7].

5. M. S. Batta, Z. Aliouat, H. Mabed, M. MERAH, “An Improved Lifetime Optimization Clustering using Kruskal’s MST and Batteries Aging for IoT Networks", The International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, July 2022 [8].
6. M. S. Batta, Z. Aliouat, H. Mabed, M. MERAH, “A Distributed Energy-Efficient Unequal Clustering based Kruskal Heuristic for IoT Networks”, In 14th International Conference on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC), Copenhagen, Danemark, 2022 [9].

Introduction générale

Contexte et Motivation

L'internet des objets (Internet of Things, IoT) a émergé comme l'une des technologies les plus transformatrices du 21^e siècle, promettant de révolutionner notre façon de vivre, de travailler et d'interagir avec notre environnement.

L'IoT repose sur des technologies sous-jacentes, dont les réseaux de capteurs sans fil (Wireless Sensors Networks, WSNs), qui jouent un rôle fondamental dans la collecte des données [10]. Elle s'appuie également sur des technologies complémentaires, telles que l'edge computing, qui assure un traitement local et efficace de ces données, tout en réduisant la dépendance aux infrastructures centralisées [11].

Avec des milliards d'appareils connectés générant des volumes massifs de données, l'IoT offre des opportunités sans précédent dans des domaines aussi variés que les villes intelligentes, l'industrie 4.0, la santé connectée et l'agriculture de précision [12, 13, 14]. Cependant, cette prolifération d'appareils soulève des défis de gestion du réseau, d'efficacité énergétique et de traitement des données [15, 16, 17].

Face à ces contraintes, le clustering constitue une solution prometteuse pour organiser efficacement ces vastes réseaux d'appareils interconnectés tout en optimisant leurs performances. En divisant les dispositifs du réseau en clusters, le clustering s'impose comme une approche fondamentale pour renforcer l'efficacité et la scalabilité des réseaux IoT. Parallèlement, l'émergence de l'apprentissage automatique (Machine Learning, ML) offre de nouvelles perspectives pour optimiser ces processus de clustering et de gestion de réseau [18, 19, 20].

Problématique

Malgré les avancées significatives dans ce domaine, plusieurs défis persistent dans l'application des techniques de ML au clustering dans les réseaux IoT :

— Il existe un manque de compréhension sur la manière d'intégrer efficacement les approches de ML (non supervisées, supervisées et par renforcement) pour répondre aux défis spécifiques du domaine.
— Certaines méthodes de clustering basées sur les approches de ML se reposent sur des paramètres d'initialisation. Elles peuvent, par exemple, nécessiter une définition préalable du nombre de clusters, ce qui n'est pas toujours réalisable dans un environnement IoT dynamique. Cette rigidité peut entraîner une mauvaise performance.
— Avec l'augmentation de la complexité et du volume des données, de nombreux systèmes IoT sont désormais gérés par des serveurs centralisés ou des infrastructures de edge computing. Cette transition soulève de nouvelles problématiques : comment optimiser la répartition du trafic entre les serveurs pour assurer une gestion uniforme et efficace dans un environnement IoT hautement dynamique ?
— Le développement d'algorithmes de clustering capables de s'adapter aux conditions variées et imprévisibles des réseaux IoT est essentiel. Ces algorithmes doivent non seulement ajuster la formation des clusters en temps réel, mais aussi être résilients face aux variations soudaines des conditions opérationnelles du réseau.

Contributions de la Thèse

Cette thèse vise à explorer et développer des méthodes innovantes de clustering pour l'IoT, basées sur des approches de ML adaptées aux contraintes des dispositifs IoT et aux limites des techniques existantes. Nos contributions s'articulent autour de trois axes principaux, chacun exploitant une approche distincte du machine learning pour répondre à des défis spécifiques des réseaux IoT :

— La première proposition se glisse sous la catégorie des approches non supervisées. Elle aborde le problème du clustering dans les réseaux WSNs en introduisant l'algorithme U-k-means, qui n'exige pas de spécifier a priori le nombre de clusters. Cette approche est combinée avec une méthode originale de sélection des CHs basée sur des algorithmes génétiques, permettant ainsi une adaptation dynamique à la topologie changeante des réseaux sans fil.
— La deuxième proposition quant à elle, utilise une approche par renforcement. Elle vise à optimiser la sélection des CHs dans un réseau WSN via une méthode basée sur le Q-learning. Cette approche permet de sélectionner de manière dynamique les CHs, en établissant un routage efficace. La fonction de récompense conçue non

seulement équilibre la charge entre les CHs, mais optimise également le débit du réseau, assure une dispersion efficace des CHs, promeut l'efficacité énergétique et minimise les retards de communication.

— Enfin la troisième proposition est basée sur une approche supervisée. Elle consiste en une méthode innovante pour optimiser l'équilibrage de la charge dans les réseaux IoT gérés par edge computing. En utilisant l'apprentissage profond, cette approche permet de prédire les variations futures du trafic et de répartir la charge de manière plus efficace entre les serveurs. L'intégration d'un algorithme génétique assure en outre une distribution optimisée de la charge, renforçant ainsi l'efficacité et la réactivité du système.

Organisation de la Thèse

Dans cette thèse, certains chapitres sont la transcription d'articles publiés dans des revues scientifiques ou conférences. Le manuscrit est structuré en deux grandes parties. La première partie présente une vue d'ensemble du domaine ainsi que l'état de l'art, tandis que la deuxième partie est consacrée aux contributions de cette recherche.

Partie I

Cette première partie offre une analyse des concepts clés liés à l'IoT et au machine learning, ainsi qu'une revue des approches existantes de clustering dans les réseaux IoT. Elle est structurée en trois chapitres :

Chapitre 1 Ce chapitre introduit le concept de l'IoT, en présentant son architecture, la typologie de ses dispositifs, ses principales applications, ainsi que les défis majeurs associés. Il explore également le concept de clustering dans les réseaux IoT, en détaillant les méthodologies principales utilisées pour clusteriser les nœuds. De plus, ce chapitre met en lumière les avantages du clustering pour relever les défis de l'IoT et propose une introduction aux différentes catégories de protocoles de clustering, qu'ils soient traditionnels, bio-inspirés ou basés sur le machine learning, en soulignant leurs limites et avantages respectifs.

Chapitre 2 Ce chapitre définit le ML et présente ses principaux paradigmes, à savoir l'apprentissage supervisé, non supervisé et par renforcement. Un focus particulier est mis sur l'apprentissage profond et son impact sur les différents paradigmes du ML. Ce chapitre introduit également les algorithmes de ML utilisés dans les proto-

coles de clustering basés sur le ML dans l'IoT, qui seront analysés dans le chapitre consacré à l'état de l'art. Enfin, ce chapitre présente différentes applications du ML.

Chapitre 3 Ce chapitre présente l'état de l'art. Il établit une taxonomie des protocoles de clustering basées sur le ML dans l'IoT. Une revue des différentes méthodes existantes, incluant à la fois des algorithmes basés sur une approche non supervisée, supervisée ou par renforcement, est présentée. Un bilan critique de ces approches est ensuite dressé, soulignant leurs avantages et leurs limites dans le contexte des réseaux IoT.

Partie II

La deuxième partie de cette thèse présente les contributions originales de cette recherche, qui se déclinent autour de trois approches basées sur le machine learning pour le clustering dans les réseaux IoT : une approche non supervisée, une approche supervisée et une approche par renforcement. Chacune est appliquée pour traiter un problème spécifique, à savoir, le clustering, la sélection des CHs et l'équilibrage de la charge. Cette partie est divisée en trois chapitres :

Chapitre 4 Ce chapitre introduit notre première contribution, qui propose une méthode de clustering des nœuds dans les WSNs en combinant l'algorithme U-k-means pour le clustering avec une approche génétique pour la sélection des CHs.

Chapitre 5 Ce chapitre présente notre deuxième contribution, qui consiste en une approche de sélection des CHs basée sur le Q-learning. Cette approche permet d'optimiser le routage tout en augmentant la performance du réseau.

Chapitre 6 Ce dernier chapitre développe notre troisième contribution, une méthode d'équilibrage dynamique de la charge dans les environnements de edge computing de l'IoT. Cette approche combine l'apprentissage profond et des algorithmes génétiques pour anticiper les variations de trafic et optimiser la répartition de la charge.

Nous terminons le manuscrit par une conclusion générale qui résume les principaux résultats obtenus au cours de cette recherche et explore les perspectives pour des travaux futurs.

PARTIE I : Vue d'ensemble et état de l'art

Chapitre 1

Internet of Things

1.1 Introduction

Dans l'ère technologique actuelle, où la connectivité façonne notre quotidien et révolutionne les industries, l'IoT émerge comme une force transformatrice, redéfinissant la manière dont nous interagissons avec le monde qui nous entoure. Plus qu'une simple innovation technologique, l'IoT représente un changement de paradigme dans notre façon de concevoir et d'utiliser les objets du quotidien, en les transformant en sources intelligentes de données et d'interactions [21].

C'est en 1999 que Kevin Ashton a posé les fondements de ce domaine en introduisant le terme « Internet of Things ». Sa vision d'un monde où les objets physiques seraient connectés à Internet et capables de communiquer entre eux a ouvert la voie à une révolution technologique sans précédent.

Aujourd'hui, l'adoption des technologies IoT connaît une croissance exponentielle. Selon des estimations récentes, des milliards d'objets seront connectés à Internet dans les prochaines années [22].

Avec cette prolifération de dispositifs connectés générant des volumes massifs de données, le clustering devient une approche essentielle pour gérer cette complexité croissante. Le clustering dans l'IoT représente bien plus qu'une simple technique d'organisation : il s'agit d'un paradigme fondamental qui transforme la manière dont nous concevons, déployons et gérons les réseaux d'objets connectés.

C'est en 2000 que Wendi B. Heinzelman *et al.* ont introduit le concept de clustering pour les réseaux de capteurs sans fil, posant ainsi les bases du clustering dans l'IoT. Leur protocole LEACH (Low-Energy Adaptive Clustering Hierarchy) a marqué un tournant décisif en proposant une approche novatrice pour l'optimisation énergétique et la gestion

efficace des réseaux de capteurs [23]. Depuis lors, l'adoption des techniques de clustering dans divers domaines de l'IoT n'a cessé de croître.

Ce chapitre présente une vue d'ensemble du paradigme de l'IoT en abordant sa définition, son architecture, sa typologie, ses applications variées et les défis auxquels il est confronté. Il introduit ensuite les concepts de base du clustering et examine les différentes catégories de protocoles de clustering, en détaillant leurs avantages et inconvénients.

1.2 Définition de l'IoT

L'IoT est un concept largement utilisé, mais sa définition varie selon les contextes, ce qui en fait souvent un terme générique. Cette absence de définition standardisée limite parfois les avancées technologiques et scientifiques, créant des perceptions différentes qui doivent être adaptées pour relever les défis variés de ce domaine. Afin de clarifier cette question, les chercheurs ont proposé plusieurs définitions, dont les suivantes :

« L'internet of Things est le réseau d'objets physiques qui contiennent une technologie intégrée pour communiquer et détecter ou interagir avec leurs états internes ou l'environnement externe. » (Groupe Gartner) [24]

« L'internet of Things est un système interconnecté d'objets physiques distinctement adressables, dotés de capacités de traitement, de détection et d'actionnement plus ou moins importantes, qui partagent la capacité d'interopérer et de communiquer par l'intermédiaire de l'internet, leur plateforme commune. » (Miraz, Mahdi H., et al.) [25]

« L'IoT, ou Internet of Things, désigne le processus de connexion d'objets physiques à Internet, des objets du quotidien tels que les ampoules, aux dispositifs médicaux, appareils portables, appareils intelligents ou encore feux de circulation routière dans les villes intelligentes. » (Red Hat) [26]

Bien que les interprétations puissent différer selon les contextes et les domaines d'application, l'essence de l'IoT est largement comprise comme l'interconnexion d'objets physiques à Internet, permettant ainsi la collecte et l'échange de données entre ces objets et d'autres systèmes. Cette interconnexion transforme des objets du quotidien en dispositifs intelligents capables de communiquer et d'interagir sans intervention humaine directe.

1.3 Architecture de l'IoT

L'architecture typique de l'IoT se structure autour de trois couches essentielles [27], à savoir la couche de perception, la couche réseau et la couche application, comme l'illustre la figure 1.1. Ces couches jouent chacune un rôle spécifique et complémentaire dans le fonctionnement global de l'IoT, assurant ainsi une gestion efficace des flux d'informations, depuis la collecte des données jusqu'à leur utilisation par les applications finales.

Couche perception constitue la première étape du processus IoT. Elle est responsable de l'interaction directe avec le monde physique à travers une multitude de dispositifs intelligents, tels que les capteurs, les actionneurs et les lecteurs RFID (Radio Frequency Identification). Ces dispositifs ont pour rôle de capturer des informations provenant de l'environnement et de convertir ces données physiques en signaux numériques exploitables. Cette couche est cruciale, car elle représente le point de départ de la chaîne de valeur de l'IoT, en collectant les données brutes nécessaires à toute prise de décision ultérieure.

Couche réseau intervient après la collecte des données et a pour mission de garantir une transmission fiable de celles-ci vers les infrastructures de traitement. Cette couche comprend une variété de technologies de communication, allant des protocoles sans fil à faible consommation d'énergie (tels que LoRa, Zigbee ou NB-IoT) aux réseaux plus classiques comme le Wi-Fi, le Bluetooth ou même les réseaux cellulaires 5G. Outre la transmission, cette couche peut également effectuer un prétraitement des données pour alléger la charge des systèmes en aval. Elle sert de pont entre les dispositifs IoT et les centres de traitement ou d'analyse des données, qu'il s'agisse de serveurs centralisés, de passerelles edge ou du cloud.

Couche application est l'étape finale, où les données, désormais prétraitées et transmises, sont exploitées pour fournir des services aux utilisateurs finaux. Cette couche englobe une large gamme d'applications IoT dans divers secteurs. Elle transforme les données collectées en informations utiles et en services concrets. Les services fournis par cette couche sont souvent personnalisés et spécifiques aux besoins des utilisateurs, et incluent des interfaces utilisateur pour faciliter l'interaction avec le système IoT.

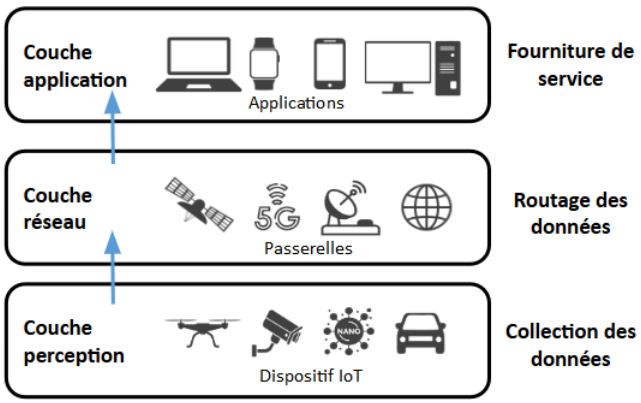


FIGURE 1.1 – Architecture de l'IoT.

1.4 Typologie de l'IoT

L'IoT englobe une vaste gamme de dispositifs et de technologies, déployés dans divers contextes pour répondre à des besoins spécifiques. En fonction de leur application et de leur environnement, les dispositifs de l'IoT peuvent être classés en plusieurs catégories, comme le démontre la figure 1.2.

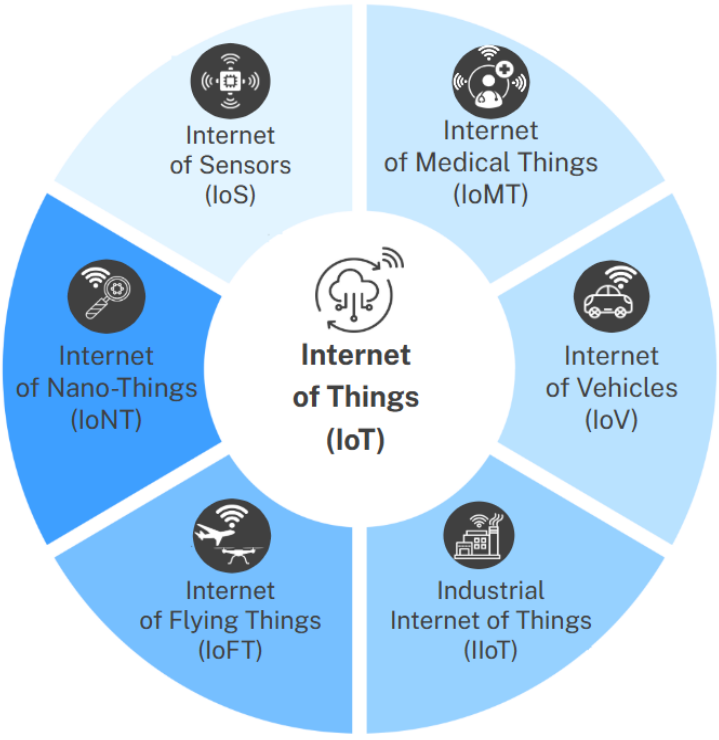


FIGURE 1.2 – Typologie de l'IoT.

1.4.1 Internet of Sensors

L'IoS (WSN) [28], est un type de réseau né de la convergence de l'électronique numérique, de la miniaturisation et des communications sans fil. Ce réseau, au cœur de l'IoT, relie les environnements physique et numérique à l'aide d'un ensemble de mini-appareils déployés dans une région d'intérêt spécifique. Ces dispositifs, dotés de capteurs ainsi que de capacités de traitement et de communication, interagissent avec l'environnement en détectant des paramètres physiques et en collectant et transmettant des données environnementales ou contextuelles. Le phénomène de l'IoS s'étend entre les réseaux terrestres, souterrains et sous-marins. Ces extensions contribuent à la mise en œuvre d'une variété d'applications.

1.4.2 Internet of Medical Things

L'IoMT [29] est un réseau de capteurs à usage spécifique conçu pour fonctionner de manière autonome en connectant différents capteurs et équipements médicaux à l'intérieur et à l'extérieur du corps humain. Ces dispositifs sont dédiés à la surveillance, au diagnostic et au traitement de la santé.

1.4.3 Internet of Vehicles

L'IoV [30] désigne un type particulier de réseau comprenant différentes entités, allant des véhicules aux infrastructures routières. Ces entités échangent des données en temps réel de manière efficace et sécurisée.

1.4.4 Industrial Internet of Things

L'IIoT [31] désigne l'utilisation de l'IoT dans les secteurs industriels. Cela se caractérise par l'intégration de capteurs, d'actionneurs et de systèmes de communication avancés dans les environnements industriels.

1.4.5 Internet of Flying Things

L'IoFT [32] se distingue par un réseau de dispositifs volants, comme les drones. Ce concept étend l'IoT à l'espace aérien, permettant la collecte et la transmission de données en temps réel pour des utilisations variées.

1.4.6 Internet of Nano-Things

L'IoNT [33] est un réseau qui a émergé grâce à l'essor de la miniaturisation. Il se compose de dispositifs à l'échelle nanométrique capables de communiquer et de réaliser des tâches spécifiques.

1.5 Application de l'IoT

Les applications de l'IoT sont vastes et couvrent de nombreux domaines, offrant des solutions innovantes et pratiques pour améliorer divers aspects de la vie quotidienne et des secteurs industriels.

Maison intelligente : L'IoT domestique fait référence aux appareils connectés, tels que les thermostats, les éclairages, les caméras de sécurité et les appareils électroménagers, qui permettent une gestion plus efficace et personnalisée de l'habitat, offrant confort, sécurité et économies d'énergie grâce aux capteurs (IoS).

Santé connectée : Les dispositifs médicaux portables, tels que les montres intelligentes et les capteurs de santé, permettent une surveillance continue des paramètres de santé, facilitant la détection précoce des anomalies et la gestion des maladies chroniques grâce aux technologies de l'Internet des objets médicaux (IoMT).

Transport intelligent : Les informations routières en temps réel sont collectées par les réseaux de véhicules connectés (IoV) pour alimenter les modèles de transport et alerter les conducteurs des problèmes de congestion et de circulation. Les drones (IoFT) peuvent être utilisés pour la surveillance du trafic et la gestion des situations d'urgence.

Villes intelligentes : Les technologies IoT sont utilisées pour améliorer les services publics tels que la gestion du trafic (IoFT), l'éclairage public, la gestion des déchets et la surveillance de l'environnement (IoS), rendant les villes plus efficaces et agréables à vivre.

Industrie 4.0 : Dans le secteur manufacturier, l'IIoT permet l'automatisation et l'optimisation des processus de production, la maintenance prédictive des machines et une meilleure gestion des chaînes d'approvisionnement.

Agriculture intelligente : Les capteurs IoT (IoS) surveillent les conditions du sol, les niveaux d'humidité et les besoins en nutriments, permettant une agriculture de précision qui améliore les rendements et réduit l'usage des ressources. L'IoNT offre

une surveillance précise à un niveau microscopique, optimisant ainsi la gestion des cultures, la détection précoce des maladies et l'utilisation efficace des pesticides et des engrais.

1.6 Défis de l'IoT

Malgré son potentiel révolutionnaire, l'IoT fait face à plusieurs défis majeurs qui doivent être relevés pour assurer son plein potentiel et son adoption à grande échelle.

Sécurité Avec des milliards d'appareils connectés, la sécurité des données et la protection de la vie privée deviennent des préoccupations majeures. Les dispositifs IoT sont souvent vulnérables aux cyberattaques en raison de leurs capacités de sécurité limitées [27]. La protection des données personnelles et la confidentialité sont également des préoccupations majeures, notamment dans les applications qui collectent des informations sensibles sur la vie privée des individus. La confidentialité de ces données doit être assurée pour éviter tout usage abusif ou non autorisé.

Interopérabilité Les différents appareils et systèmes IoT doivent être capables de communiquer et de fonctionner ensemble de manière transparente. L'absence de normes universelles pose des problèmes d'interopérabilité entre les dispositifs de différents fabricants [34]. Ce manque de standardisation peut freiner l'innovation et limiter l'efficacité des solutions IoT.

Gestion de l'énergie De nombreux appareils IoT sont alimentés par des batteries et doivent fonctionner pendant de longues périodes sans recharge fréquente. L'efficacité énergétique, l'optimisation de la durée de vie des batteries et le développement de solutions d'alimentation alternatives sont cruciaux pour assurer la viabilité des solutions IoT [35].

Scalabilité Les infrastructures IoT doivent être capables de s'adapter à une croissance exponentielle du nombre d'appareils connectés sans compromettre la performance et la fiabilité.

Gestion des données La quantité massive de données générée par les dispositifs IoT nécessite des solutions efficaces pour la collecte, le stockage, le traitement et l'analyse des données en temps réel [36].

1.7 Clustering dans l'IoT

Dans le contexte des réseaux sans fil, le clustering est une technique de gestion de réseau visant à améliorer l'efficacité de la communication en organisant les nœuds du réseau en clusters ou groupes. Les noeuds sont groupés en fonction de critères tels que la similitude, proximité géographique, ou rôles spécifiques [37]. Ces clusters sont formés pour faciliter la coordination et la communication entre les nœuds, réduire la latence et la charge du réseau, et prolonger la durée de vie de la batterie des appareils.

Comme le démontre la figure 1.3, les deux principaux acteurs dans un réseau clusterisé sont les Cluster Heads (CHs) et les membres du groupe.

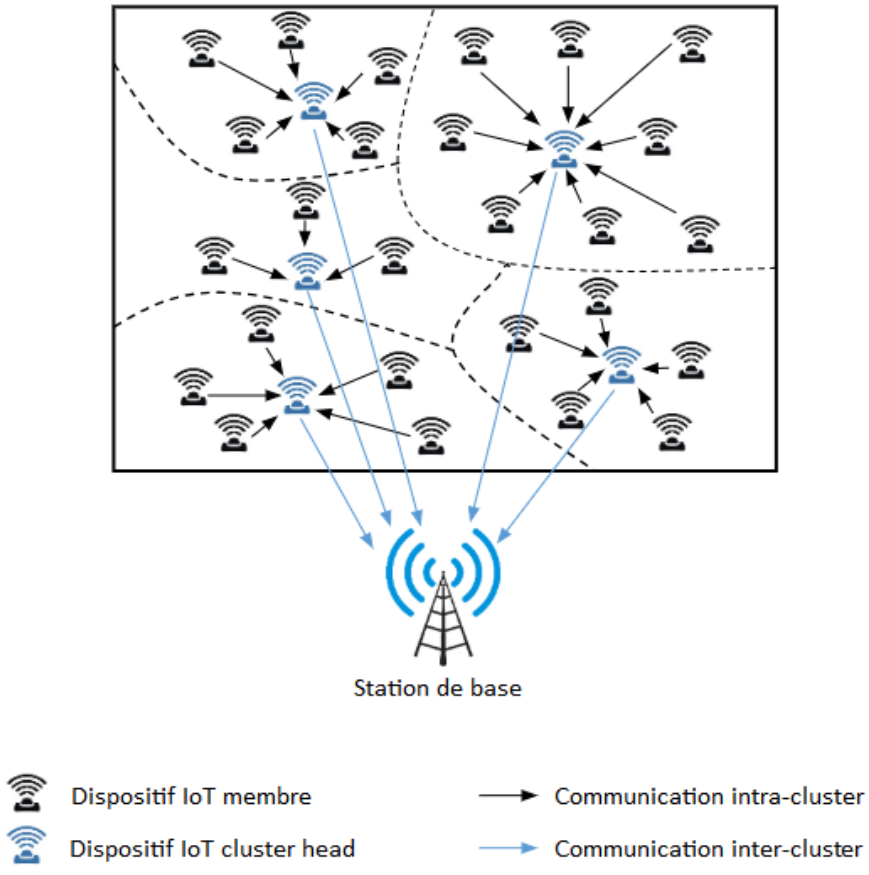


FIGURE 1.3 – Réseau IoT en clusters.

Cluster Heads : Les chefs de groupe sont des dispositifs sélectionnés pour agir en tant que points de coordination et de contrôle au sein des clusters. Ils sont responsables de la gestion des activités au sein de leur cluster, y compris la coordination des communications entre les membres du groupe. Les CHs peuvent être sélectionnés

en fonction de divers critères, tels que leur niveau de batterie, leur puissance de transmission ou leur position géographique, afin d'optimiser les performances du réseau.

Membres du groupe : Ce sont les nœuds qui font partie d'un cluster et qui communiquent généralement avec le chef de groupe pour transmettre des données ou recevoir des instructions. Les membres du groupe peuvent être des capteurs, des dispositifs IoT, des appareils mobiles ou d'autres nœuds du réseau sans fil.

Comme le démontre la figure 1.3 ces acteurs coopèrent afin d'établir un routage efficace. Les membres collectent les données et les CHs les transmettent à la station de base.

1.7.1 Méthodologie

Cette section présente la méthodologie à suivre pour réaliser le clustering dans l'IoT.

Collection des informations réseau

La première étape consiste à collecter les données pertinentes des dispositifs IoT. Ces données peuvent inclure : La localisation géographique, le niveau d'énergie, les capacités de traitement et de stockage, et les types de capteurs. La collecte des données peut se faire selon plusieurs approches :

Approche centralisée Dans cette approche, toutes les données des dispositifs IoT sont envoyées à un point central, par exemple une station de base (Base Station (BS) en anglais), pour le traitement et le stockage. Cette approche offre une vue globale des données mais peut présenter des défis en termes de latence et de bande passante pour les grands réseaux IoT.

Approche distribuée Cette approche implique le traitement et le stockage des données au niveau local ou en périphérie du réseau. Elle peut réduire la latence et la charge sur le réseau principal, mais nécessite une gestion plus complexe des données distribuées.

Approches de clustering

L'objectif principal d'un protocole de clustering est l'organisation des dispositifs réseau en groupe, ceci peut se faire selon différentes approches :

Approche centralisé Dans cette approche, un nœud central, une station de base, ou un serveur est responsable de la formation et de la gestion des clusters. Ainsi toutes les données des dispositifs IoT sont envoyées à ce point central. Ce nœud central effectue l'analyse et le clustering. Cette approche offre une vue globale des données mais peut présenter des défis en termes de latence et de bande passante pour les grands réseaux IoT.

Approche distribué Dans ce modèle, les dispositifs IoT participent activement au processus de formation des clusters. Les décisions de clustering sont prises localement ou de manière collaborative entre les dispositifs. Cette approche peut réduire la latence et la charge sur le réseau, mais nécessite une gestion plus complexe des données distribuées et peut être plus complexe à mettre en œuvre.

Approche semi-centralisé Cette approche hybride combine des éléments des deux approches centralisées et distribuées. Par exemple, la sélection des CHs peut se faire de manière centralisée, tandis que la formation des clusters et la gestion intra-cluster sont réalisées de manière distribuée. Cette méthode offre un équilibre entre la vue globale du réseau avec une gestion locale efficace.

Stratégies de clustering

Indépendamment de l'approche centralisée ou distribuée, différentes stratégies de clustering peuvent être appliquées :

Clustering basé sur la sélection des CHs Dans cette approche, les CHs sont d'abord sélectionnés parmi les nœuds du réseau en fonction de critères prédéfinis. Une fois les CHs sélectionnés, les autres nœuds du réseau sont assignés à des clusters en fonction de leur proximité géographique ou d'autres facteurs, avec les CHs agissant comme points de référence ou de coordination.

Clustering basé sur les groupes Cette approche commence par la formation de groupes de nœuds en fonction de différents critères tels que la proximité géographique ou la puissance de transmission. Ensuite, un CH est sélectionné au sein de chaque groupe formé, généralement en fonction de la capacité ou de la disponibilité de ressources. Une fois que les chefs de groupe sont désignés, les clusters sont formés autour d'eux en regroupant les membres du groupe en fonction de leur proximité ou d'autres facteurs.

Choix de l'algorithme de clustering

Le choix de l'algorithme de clustering est une étape cruciale qui dépend de plusieurs facteurs, à savoir : la nature des données, l'objectif du clustering, et les contraintes computationnelles, ainsi que l'approche et la stratégie de clustering adoptée.

Maintenance et adaptation dynamique

Il est nécessaire de mettre en place des mécanismes pour maintenir et adapter dynamiquement les clusters en fonction des changements dans le réseau IoT, tel que l'arrivé de nouveaux dispositifs ou la déconnexion de certains. Ces mécanismes doivent être capables d'évaluer en temps réel la topologie du réseau, d'analyser la performance des clusters existants et de réagir rapidement aux variations d'activité afin de maintenir la performance du réseau.

1.8 Les avantages du clustering pour relever les défis de l'IoT

Le clustering est considéré comme la méthode la plus efficace pour gérer les systèmes IoT, car il permet de collecter les données IoT de manière efficace avec un nombre minimal de communications et de diffuser les informations collectées en vue d'un traitement ultérieur. Cette technique a été largement utilisée pour relever divers défis liés à l'IoT. Les défis les plus fréquents sont examinés ci-dessous.

Sécurité La sécurité des utilisateurs, des communications, des données, des services, des réseaux et des équipements continue d'être un problème critique dans l'Internet d'aujourd'hui. Avec l'émergence de l'IoT, l'ampleur du problème prend une autre dimension, car des milliers d'objets contraints sont connectés en permanence à l'Internet et intégrés dans notre vie quotidienne. Ils risquent donc d'être la cible des menaces Internet classiques et pourraient même faire émerger de nouvelles générations d'attaques. De plus, notre utilisation quotidienne révèle nos habitudes, notre état de santé, notre localisation géographique, et d'autres informations privées [38]. Il est strictement nécessaire de garantir la confidentialité et la vie privée des données que l'utilisateur considère comme sensibles. Par conséquent, les dispositifs, les transmissions et le stockage des données IoT doivent être soigneusement contrôlés et sécurisés.

Les mécanismes de clustering peuvent être utilisés pour surveiller le comportement des dispositifs au sein d'un cluster. En identifiant rapidement les comportements anormaux ou suspects, il est possible de détecter des intrusions ou des attaques potentielles. De plus, en utilisant des CHs pour gérer la communication au sein de chaque groupe, il est possible de contrôler plus efficacement les accès aux dispositifs et aux données. Cela permet d'appliquer des politiques de sécurité spécifiques à chaque cluster, renforçant ainsi la sécurité globale. Enfin, en limitant le nombre de dispositifs directement connectés au réseau principal et en les regroupant dans des clusters, le nombre de points d'entrée pour une éventuelle attaque est réduit, ce qui diminue les risques d'accès non autorisé.

L'interopérabilité Le clustering permet d'établir des normes communes pour la communication entre différents dispositifs. En regroupant des appareils similaires, il devient plus facile de créer des protocoles de communication standardisés qui facilitent l'interaction entre les systèmes hétérogènes. Cela permet aux dispositifs de différents fabricants de fonctionner ensemble sans nécessiter d'adaptations spécifiques.

Bien que le clustering puisse offrir des solutions prometteuses pour améliorer l'interopérabilité dans l'IoT, il nécessite une attention particulière aux défis associés à la diversité des dispositifs et aux exigences en matière de sécurité. En continuant à développer et à affiner les techniques de clustering, il sera possible d'améliorer l'intégration et la communication entre les systèmes IoT, favorisant ainsi un écosystème plus interconnecté et efficace.

Gestion de l'énergie Les capteurs sont limités en termes de capacité de calcul, de stockage de mémoire et d'énergie. Étant souvent alimentés par des batteries ou des sources d'énergie limitées, la gestion de la consommation d'énergie est l'aspect le plus crucial pour garantir leur fonctionnement optimal et prolonger leur durée de vie. Par conséquent, les solutions en matière de protocoles de communication, de protocoles de sécurité et de technologies de transmission pour l'IoT doivent tenir compte de ces limitations. En effet, une consommation efficace des ressources est nécessaire pour prendre en charge les diverses contraintes des protocoles complexes et gourmands en mémoire.

Le clustering tente de minimiser la dissipation globale d'énergie dans le réseau, ce qui prolonge systématiquement la durée de vie des appareils et équilibre la consommation d'énergie entre les différents nœuds IoT. Certaines solutions de clustering s'appuient sur l'équilibrage des rôles pour empêcher certains appareils de

gaspiller rapidement de l'énergie par rapport aux autres nœuds du réseau. En outre, de nombreux protocoles fonctionnent par cycles, dans lesquels un CH différent est élu à chaque cycle, et qui pourrait suivre une politique de rotation prenant en compte le niveau d'énergie des nœuds au sein d'un cluster pour déterminer le meilleur CH [39]. La rotation dynamique évite le fonctionnement répétitif des mêmes CH et réduit ainsi leur usure prématurée [40]. En outre, des solutions de clustering peuvent être adoptées pour réduire la taille et le nombre de paquets envoyés à la station de base [41].

Scalabilité Nous nous attendons à ce que des milliards de dispositifs IoT peuplent l'Internet du futur. Il est donc fortement recommandé d'adopter de nouveaux mécanismes qui soutiennent efficacement l'évolutivité continue du nombre d'objets connectés. Le nombre d'appareils IoT déployés dans l'environnement peut varier de quelques centaines à quelques milliers, voire plus, en fonction de l'application. Cisco estime que 50 milliards d'appareils sont actuellement connectés à l'Internet [42].

Dans un système clusterisé, il est plus facile d'intégrer de nouveaux capteurs ou dispositifs. Les nouveaux dispositifs peuvent être ajoutés à un cluster existant sans perturber le reste du réseau, facilitant ainsi l'expansion du système. Le clustering garantit ainsi l'évolutivité du réseau en localisant la transmission des messages, en réduisant l'encombrement et en améliorant les performances du réseau.

Gestion des données Les milliards d'objets connectés génèrent d'énormes volumes de données. Le transfert efficace de ces données, depuis leur source jusqu'à leur destination, est essentiel pour garantir le bon fonctionnement des systèmes de communication et des applications basées sur l'IoT.

Grâce au clustering, les CHs du réseau peuvent regrouper les données avant de les transmettre à un serveur central ou à un système de traitement, ce qui réduit le volume de données à transférer et optimise ainsi la bande passante du réseau [41].

1.9 Introduction aux protocoles de clustering dans l'IoT

Le clustering dans l'IoT est essentiel pour gérer efficacement les nombreux appareils connectés, optimiser les communications et améliorer l'efficacité énergétique. Cette section explore trois approches principales : traditionnelle, bio-inspirée et basée sur le ML.

Le tableau 1.1 présente une vue d'ensemble comparative de ces trois approches selon différents critères essentiels pour les réseaux IoT.

1.9.1 Protocoles Traditionnels

Les méthodes traditionnelles de clustering dans l'IoT reposent sur des algorithmes déterministes, souvent basés sur des métriques de distance, d'énergie ou de capacité de calcul des nœuds. Les protocoles classiques, comme LEACH [23], ont été pionniers dans l'optimisation énergétique des réseaux IoT. LEACH, introduit par Heinzelman et al. en 2000, organise les capteurs en clusters pour minimiser la consommation d'énergie. Il sélectionne périodiquement des CHs de manière aléatoire pour équilibrer la charge énergétique entre les nœuds, ces derniers étant responsables de l'agrégation des données et de leur transmission à la station de base. Les nœuds non-leaders envoient leurs données aux leaders, réduisant ainsi les communications directes coûteuses en énergie avec la station de base. Différentes variantes de LEACH ont été proposées, comme HEED (Hybrid Energy-Efficient Distributed clustering approach) [43]. Ces travaux ont établi des bases solides pour le développement de solutions efficaces dans ce domaine.

1.9.2 Protocoles Bio-inspirés

Le clustering bio-inspiré s'inspire des mécanismes naturels et des comportements des organismes vivants pour concevoir des algorithmes de clustering. Ces méthodes sont souvent utilisées pour optimiser le clustering en s'adaptant dynamiquement aux variations des données. Parmi les approches bio-inspirées, les algorithmes génétiques (Genetic Algorithm, GA), l'algorithme d'optimisation des colonies de fourmies (Ant Colony Optimization, ACO) et l'algorithme de l'optimisation par essaims particulaires (Particle Swarm Optimization, PSO) se distinguent par leur efficacité [44]. Les algorithmes génétiques imitent le processus de sélection naturelle pour faire évoluer des populations de solutions vers des résultats optimaux. Ils utilisent des gènes qui peuvent représenter les identités des CHs et évoluent par sélection, mutation et croisement afin de trouver un ensemble optimal qui minimise la consommation d'énergie et optimise la couverture du réseau. L'ACO, quant à lui, s'inspire du comportement des fourmis, où des phéromones influencent le choix de chemin lors de la recherche de solutions. Dans le contexte du clustering, les nœuds du réseau agissent comme des fourmis, déposent des "marques" numériques (similaires aux phéromones), et évaluent les chemins vers les CHs potentiels. Les chemins avec la meilleure qualité, déterminée par des critères comme la distance ou la

consommation d'énergie, attirent davantage de "fourmis" et, avec le temps, un ensemble optimal de CHs et de clusters se forme, équilibrant efficacement la charge du réseau et minimisant la consommation d'énergie. Enfin, l'algorithme PSO simule le comportement collectif d'un groupe d'oiseaux ou de poissons pour trouver des solutions optimales. PSO peut effectuer le clustering en modélisant chaque nœud comme une particule dans un espace de solution. Chaque particule représente une configuration possible de CHs et se déplace dans l'espace de solution en ajustant sa position en fonction de sa propre expérience (la meilleure solution qu'elle a trouvée) et de l'expérience collective de l'essaim (la meilleure solution globale). Au fil des itérations, les particules convergent vers une solution optimale, où les CHs sont sélectionnés de manière à minimiser la consommation d'énergie et à optimiser la distribution des nœuds, améliorant ainsi la performance globale du réseau IoT.

1.9.3 Protocoles basés sur Machine Learning

Le clustering basé sur ML est une approche récente et innovante qui utilise des algorithmes pour apprendre des schémas dans les données et en extraire des groupes. Contrairement aux méthodes traditionnelles, ces techniques peuvent s'adapter en temps réel aux conditions changeantes. Les protocoles basés sur ML exploitent ses différents paradigmes pour automatiser la formation de clusters, à savoir, l'apprentissage non supervisé, l'apprentissage supervisé et l'apprentissage par renforcement [3]. L'apprentissage non supervisé est souvent utilisé pour découvrir des relations dans des ensembles de données sans étiquettes. Il peut être appliqué pour regrouper les dispositifs IoT. L'apprentissage supervisé, quant à lui, nécessite des données étiquetées pour entraîner les modèles. Il peut améliorer la précision du clustering en apprenant à partir d'exemples. Enfin, l'apprentissage par renforcement est une approche où un agent apprend à prendre des décisions en interagissant avec un environnement. Cette méthode peut être appliquée au clustering en permettant aux systèmes d'ajuster dynamiquement leur stratégie de clustering en fonction des retours d'expérience.

Le choix de l'approche de clustering dans l'IoT dépend des besoins spécifiques du réseau, de sa taille et des ressources disponibles. Comme le démontre le tableau 1.1, le clustering traditionnel offre simplicité et facilité de mise en œuvre, ce qui le rend idéal pour les petits réseaux statiques. Les approches bio-inspirées, quant à elles, fournissent un bon équilibre entre adaptabilité et complexité, convenant aux réseaux de taille moyenne dans des conditions changeantes. En revanche, le clustering basé sur le ML, bien que plus complexe et coûteux, offre la meilleure adaptabilité et efficacité, étant

TABLE 1.1 – Comparaison des protocoles de clustering dans l'IoT (traditionnel, bio-inspiré, et basé sur ML)

Critère	Traditionnel	Bio-Inspiré	Basé Machine Learning
Complexité de mise en œuvre	Faible	Moyenne	Élevée
Adaptabilité	Limitée	Bonne	Excellente
Efficacité énergétique	Variable	Généralement efficace	Potentiellement très efficace
Scalabilité	Limitée	Bonne	Excellente
Flexibilité	Faible	Moyenne	Élevée
Capacité d'optimisation	Statique	Dynamique	Continue et prédictive
Coût de déploiement	Faible	Moyen	Élevé
Gestion de la complexité des données	Basique	Modérée	Avancée

particulièrement adapté aux réseaux hautement dynamiques et aux scénarios complexes.

1.10 Conclusion

En explorant les aspects essentiels de l'IoT, ce chapitre nous a permis de fournir une compréhension de la nature, des avantages et des obstacles de cette technologie révolutionnaire. Les défis associés à cette technologie soulignent la nécessité de solutions innovantes et robustes pour optimiser son fonctionnement. C'est dans ce contexte que le concept de clustering des réseaux IoT émerge comme une approche prometteuse pour exploiter pleinement le potentiel des réseaux IoT et surmonter certains de leurs défis.

Le clustering dans l'IoT est une approche cruciale pour organiser les dispositifs IoT. Nous avons exploré différentes méthodes de clustering, notamment les approches traditionnelles, bio-inspirées et basées sur le ML. Chacune de ces techniques offre des avantages distincts et répond à des besoins spécifiques dans le contexte de l'IoT. Dans le reste du manuscrit, nous nous concentrerons plus particulièrement sur le clustering basé sur le ML. Cette approche en plein essor peut apporter une intelligence adaptative au clustering dans l'IoT en permettant aux réseaux de s'auto-organiser en fonction des variations dynamiques du trafic et de la topologie. Les algorithmes apprennent à anticiper les besoins en ressources et à ajuster la formation des clusters en temps réel, optimisant ainsi la consommation d'énergie et la qualité des transmissions.

Dans le prochain chapitre, nous introduirons les concepts du ML et discuterons de ses algorithmes courants et de leurs applications potentielles.

Chapitre 2

Machine Learning

2.1 Introduction

Dans l'ère numérique actuelle, où les données sont omniprésentes et les capacités de calcul ne cessent de croître, le ML émerge comme une technologie transformative. Ce domaine fascinant de l'intelligence artificielle permet aux ordinateurs d'apprendre et de s'améliorer à partir de l'expérience, sans être explicitement programmés pour chaque tâche spécifique.

C'est au cours de l'année 1950 qu'Alan Turing pose les fondements de ce domaine, ouvrant la voie à la réflexion sur les machines pensantes [45]. Quelques années plus tard, en 1959, Arthur Samuel introduit le terme « machine learning » dans son travail pionnier sur un programme apprenant à jouer aux dames [46]. Depuis lors, le ML a connu des avancées significatives, permettant la résolution de problèmes complexes dans divers domaines.

Dans ce chapitre, nous commencerons par définir le ML, avant d'examiner ses principaux types, notamment l'apprentissage supervisé, non supervisé et par renforcement. Nous mettrons ensuite en lumière l'apprentissage profond et introduirons le concept de clustering en ML. Quelques algorithmes clés seront également présentés. Enfin, nous explorerons les nombreuses applications de cette technologie.

2.2 Définition du ML

Le concept de machine learning est omniprésent dans le paysage technologique actuel, mais sa signification peut varier considérablement en fonction des applications, ce qui

en fait parfois un terme un peu flou. Afin de clarifier cette notion, les chercheurs ont proposé plusieurs définitions, dont les suivantes :

« On dit qu'un programme d'ordinateur apprend de l'expérience E par rapport à une catégorie de tâches T et à une mesure de performance P si sa performance pour les tâches de T, mesurée par P, s'améliore avec l'expérience E. » (Tom Mitchell)[45]

« Le machine learning (ou apprentissage automatique) est une technique qui consiste à entraîner un ordinateur à identifier des schémas, à établir des prédictions et à apprendre à partir d'expériences passées sans programmation explicite. » (Red Hat)[47]

Ces définitions, bien que formulées différemment, convergent vers l'idée centrale selon laquelle le ML implique des systèmes capables d'améliorer leurs performances à travers l'expérience, sans intervention humaine directe.

2.3 Vocabulaire du ML

Le ML repose sur un vocabulaire spécifique. Cette section met en lumière les termes fondamentaux qui sous-tendent ce domaine.

Données et caractéristiques Au cœur de tout système de ML se trouvent les données. Un ensemble de données est une collection utilisée pour entraîner et tester les modèles. Les ensembles de données doivent être représentatifs et bien équilibrés pour garantir des performances optimales.

Ces données sont généralement représentées sous forme de caractéristiques. Les caractéristiques peuvent être numériques ou catégorielles. La qualité et la pertinence des caractéristiques sont cruciales pour la performance des modèles d'apprentissage automatique [48].

Avant leur utilisation, les données doivent passer par différentes étapes, à savoir : le nettoyage des données et la suppression des redondances, le prétraitement et la normalisation, ainsi que la sélection des caractéristiques nécessaires pour l'application.

Modèles Un modèle de ML est une représentation mathématique des relations entre les variables d'entrée et de sortie, apprise à partir des données [49]. Il est utilisé pour faire des prédictions ou des classifications sur de nouvelles données.

Les modèles peuvent prendre diverses formes, des simples régressions linéaires aux réseaux de neurones complexes [50]. Le choix du modèle dépend de la nature du problème et des données disponibles.

Avant leur déploiement, les modèles doivent être évalués. L'évaluation des modèles peut s'effectuer sur la base de certaines métriques de performance ou par validation croisée.

Apprentissage L'apprentissage en ML fait référence au processus d'amélioration et d'optimisation des performances du modèle sur une tâche spécifique à travers l'expérience [51]. L'optimisation est le processus mathématique par lequel cet apprentissage se produit, généralement en minimisant une fonction de coût qui mesure l'écart entre les prédictions du modèle et les vraies valeurs.

2.4 Paradigmes du ML

Le ML comprend trois paradigmes fondamentaux, chacun adapté à différents types de problèmes et de données, comme le démontre la figure 2.1.

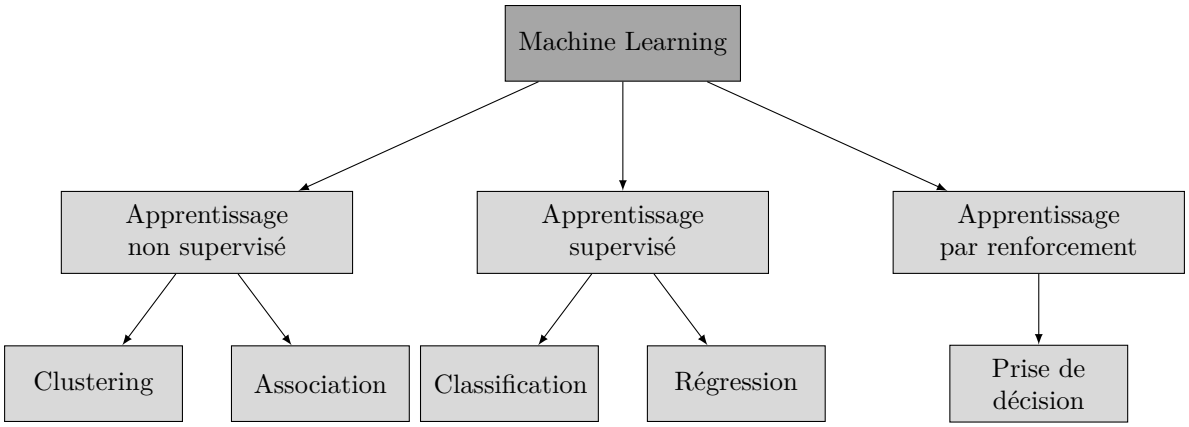


FIGURE 2.1 – Paradigmes fondamentaux du Machine Learning.

2.4.1 Apprentissage supervisé

L'apprentissage supervisé (Supervised Learning, SL) est un paradigme où les modèles sont formés à partir de données étiquetées. Cela signifie que chaque exemple d'entraînement comprend à la fois des entrées et les sorties désirées. Les modèles apprennent à établir une relation entre ces entrées et ces sorties afin de faire des prédictions sur de nouvelles données [52].

Les prédictions peuvent être catégorielles ou numériques, relevant ainsi soit de la classification, soit de la régression, selon la nature de la sortie (discrète ou continue). Les algorithmes de classification sont employés lorsque la variable de sortie est catégorielle. Une variable catégorielle représente une valeur appartenant à l'une des classes définies, comme « vrai » ou « faux ». En revanche, les algorithmes de régression sont utilisés pour prédire des variables continues lorsqu'une relation existe entre la variable d'entrée et la variable de sortie. Les variables continues sont des valeurs pouvant prendre n'importe quelle valeur au sein d'un certain intervalle.

2.4.2 Apprentissage non supervisé

L'apprentissage non supervisé (Unsupervised Learning, UL), contrairement à l'apprentissage supervisé, utilise des données non étiquetées. Dans ce cas, les modèles cherchent à identifier des structures ou des motifs dans les données sans disposer d'exemples de sortie spécifiques [52].

Les modèles d'apprentissage non supervisé permettent de découvrir des structures cachées et des informations à partir de données non étiquetées. Ces modèles peuvent être classés en deux grandes catégories de problèmes : l'association et le regroupement. L'association vise à identifier les relations entre les variables dans de grands ensembles de données, en détectant les groupes d'éléments qui apparaissent fréquemment ensemble. Le regroupement, quant à lui, est une technique qui divise les points de données en groupes appelés clusters, où les objets similaires sont regroupés, tandis que ceux qui présentent peu ou pas de similitudes se trouvent dans des groupes distincts. L'analyse des clusters met en évidence les similitudes entre les objets de données et les classe selon la présence ou l'absence de ces similitudes.

2.4.3 Apprentissage par renforcement

L'apprentissage par renforcement (Reinforcement Learning, RL) est un paradigme où un agent apprend à interagir avec un environnement en prenant des décisions séquentielles. L'agent reçoit des récompenses ou des pénalités en fonction de ses actions, ce qui lui permet d'apprendre une stratégie optimale au fil du temps [52].

L'apprentissage par renforcement traite un type spécifique de problème où les décisions sont prises de manière séquentielle pour atteindre un objectif à long terme [53], mais il peut également être adapté à d'autres types de problèmes.

2.5 Deep Learning

L'apprentissage profond (Deep Learning, DL), est un domaine de l'intelligence artificielle, inspiré des réseaux de neurones biologiques, qui vise à simuler la façon dont le cerveau humain traite les informations [54, 55].

Le réseau de neurones artificiels (Artificial Neural Networks, ANNs) est l'élément de base du DL. Comme le montre la figure 2.2, il est composé de neurones artificiels organisés en couches, interconnectées par des connexions unidirectionnelles appelées poids [55] :

Couche d'entrée où les données sont introduites dans le réseau.

Couches cachées où les neurones transforment les données en appliquant des pondérations et des fonctions d'activation.

Couche de sortie où les résultats finaux sont produits.

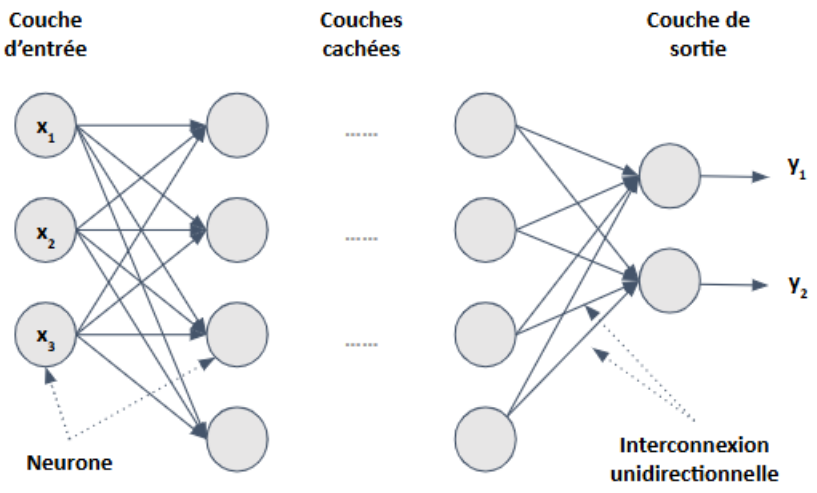


FIGURE 2.2 – Schéma de l'architecture d'un réseau de neurones artificiels composé d'une couche d'entrée, de plusieurs couches cachées et d'une couche de sortie.

La profondeur dans le DL fait référence aux nombreuses couches cachées de réseaux de neurones interconnectées. Cette profondeur permet au modèle d'apprendre des représentations de plus en plus abstraites et complexes des données [56].

Il existe plusieurs types de réseaux de neurones dans l'apprentissage profond, adaptés à différents types de données :

Réseaux de Neurones Convolutifs Les réseaux de neurones convolutifs (Convolutional neural network, CNNs) sont un type de réseau est spécialement conçu pour

traiter des données structurées en grille. Contrairement aux réseaux de neurones simples, où chaque neurone est connecté à tous les neurones de la couche précédente, ces réseaux utilisent des couches de convolution qui appliquent des filtres pour détecter des motifs locaux dans les données.

Réseaux de Neurones Récurrents Les réseaux de neurones récurrents (Recurrent Neural Networks, RNNs) sont un type de réseau se distingue par sa capacité à traiter des données séquentielles. Contrairement aux réseaux de neurones simples, où les informations sont traitées de manière indépendante, ces réseaux possèdent des boucles internes qui leur permettent de conserver une mémoire des états précédents. Cette architecture leur permet de capturer les dépendances temporelles dans les données, en reliant chaque étape de la séquence à celles qui la précèdent.

Le DL peut être intégré aux algorithmes classiques d'apprentissage automatique : supervisé, non supervisé, et par renforcement, pour améliorer leurs performances [57, 58]. En apprentissage supervisé, l'algorithme utilise des réseaux de neurones pour apprendre à partir d'exemples étiquetés et faire des prédictions sur de nouvelles données. En revanche, dans l'apprentissage non supervisé, il tente de découvrir des motifs ou des regroupements dans des données non étiquetées. Enfin, en apprentissage par renforcement, un agent utilise des réseaux de neurones profonds pour estimer des fonctions de valeur ou des politiques.

2.6 Clustering en ML

Le clustering est un concept central en ML. C'est une technique clé du ML non supervisé, utilisée pour regrouper des données en fonction de leur similarité, sans l'utilisation de labels ou d'exemples préclassifiés [59]. L'objectif du clustering est de diviser un ensemble de données en sous-ensembles appelés clusters, où les objets au sein d'un même cluster sont plus similaires entre eux qu'avec ceux des autres clusters, comme le démontre la figure 2.3. Cela permet d'identifier des structures cachées, des motifs ou des groupes d'éléments similaires dans des ensembles de données non étiquetés.

Le clustering repose sur des algorithmes qui calculent des distances ou des similarités entre les données pour les regrouper. Le principe général est de minimiser la variance intra-cluster (dissimilarité entre les points au sein d'un même cluster) tout en maximisant la variance inter-cluster (dissimilarité entre les différents clusters). Par exemple, le clustering peut être effectué en calculant la distance par rapport à des points centraux appelés centroïdes, comme illustré dans la figure 2.3.

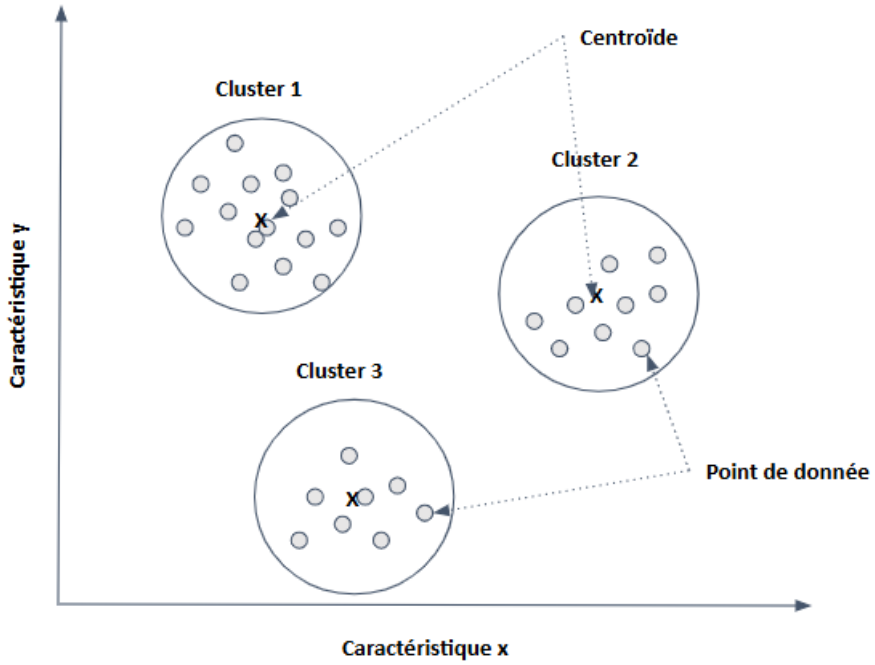


FIGURE 2.3 – Clustering d'un ensemble de données selon deux caractéristiques x et y.

Le clustering est utilisé dans de nombreux domaines pour résoudre des problèmes complexes où les relations entre les données ne sont pas explicites. Nous pouvons citer quelques exemples d'applications :

- Analyse de marché : Identifier des groupes de clients partageant des comportements d'achat similaires afin de cibler des stratégies marketing adaptées à chaque segment.
- Biologie computationnelle : Regrouper des gènes ou des protéines en fonction de leur expression dans différentes conditions pour découvrir des fonctions biologiques similaires.
- Segmentation d'images : Identifier des régions d'intérêt dans des images, par exemple en regroupant des pixels aux couleurs similaires pour détecter des objets ou des contours.
- Détection d'anomalies : Identifier des comportements inhabituels ou des points de données qui diffèrent significativement du reste des clusters, ce qui est utile dans la détection de fraudes ou de pannes.

Comme mentionné précédemment, le clustering est généralement réalisé à l'aide d'algorithmes d'apprentissage non supervisé. Cependant, il est possible de concevoir des algorithmes de clustering en utilisant des techniques supervisées ou d'apprentissage par

renforcement, permettant ainsi de tirer parti d'informations étiquetées ou de rétroactions pour améliorer le processus de clustering.

Bien que le clustering soit largement utilisé, il présente plusieurs défis, tels que le choix du nombre optimal de clusters, la gestion de grandes bases de données à haute dimension, les données non structurées et complexes, ainsi que la sensibilité aux outliers (les points de données anormalement distants des autres points). Ces défis ouvrent de nouvelles perspectives pour le clustering en ML.

2.7 Quelques algorithmes ML

Il existe une multitude d'algorithmes de ML, parmi les algorithmes inclus dans le manuscrit, sont les suivants :

2.7.1 Algorithmes de clustering

K-means L'algorithme K-means [60] est l'une des méthodes de regroupement (clustering) les plus populaires en apprentissage automatique non supervisé. Il est utilisé pour partitionner un ensemble de données en un nombre prédéfini de groupes, ou clusters, en fonction de la similarité des points de données.

K-means fonctionne en attribuant chaque point de données au cluster le plus proche, basé sur la distance euclidienne. Le processus commence par la sélection aléatoire de K centroïdes, qui représentent le centre de chaque cluster. Les étapes suivantes sont ensuite répétées jusqu'à la convergence. Chaque point de données est affecté au cluster dont le centroïde est le plus proche. Le centroïde de chaque cluster est mis à jour en calculant la moyenne de tous les points qui lui sont assignés. Le processus se poursuit jusqu'à ce que les centroïdes ne changent plus, ou que la variation entre les affectations des points soit minimale, signifiant que l'algorithme a convergé.

K-means est facilement affecté par le bruit et les valeurs aberrantes et dépend fortement de l'initialisation de ses paramètres (le nombre de clusters, les centroïdes initiaux, et la métrique de distance) pour réaliser le clustering. Plusieurs variantes ont été proposées afin de régler ces problèmes [61].

K-medoids L'algorithme K-medoids [62] est une variante robuste de l'algorithme K-means, conçu pour atténuer l'impact des points aberrants (outliers) et des données bruitées. Alors que K-means utilise la moyenne pour définir les centroïdes des

clusters, K-medoids se base sur les médianes ou des points réels appelés médoïdes, ce qui le rend moins sensible aux données extrêmes.

Comme K-means, K-medoids partitionne un ensemble de données en K clusters, mais au lieu de choisir des centroïdes calculés comme la moyenne des points d'un cluster, il sélectionne des points représentatifs de l'ensemble de données, appelés médoïdes. Un médoïde est défini comme le point dans un cluster dont la distance totale avec tous les autres points du cluster est minimale.

X-means L'algorithme X-means [63] est une extension de l'algorithme K-means qui vise à résoudre la nécessité de définir à l'avance le nombre de clusters K. Alors que K-means demande de spécifier le nombre exact de clusters, X-means détermine automatiquement ce nombre optimal en fonction des données, tout en améliorant l'efficacité du processus de regroupement.

X-means commence par appliquer l'algorithme K-means avec un petit nombre initial de clusters. Ensuite, il divise les clusters existants de manière dynamique et évalue la qualité du regroupement à l'aide d'un critère de validation, tel que le critère d'information bayesien (Bayesian Information Criterion, BIC) [64]. Si diviser un cluster améliore la qualité globale du modèle, la division est conservée. Sinon, le cluster reste inchangé. Le processus continue jusqu'à ce que l'algorithme trouve le nombre de clusters optimal.

Fuzzy C-means (FCM) L'algorithme Fuzzy C-means (FCM) [65] est une méthode de clustering qui étend l'approche traditionnelle des algorithmes de partitionnement en introduisant un degré d'appartenance flou. Contrairement aux autres algorithmes de clustering, où chaque point est assigné à un seul cluster, FCM permet à chaque point d'appartenir à plusieurs clusters avec des degrés d'appartenance différents. Cela rend FCM particulièrement adapté aux ensembles de données présentant des frontières floues entre les clusters.

Le fonctionnement de FCM repose sur l'optimisation d'une fonction objective qui cherche à minimiser la distance entre les points de données et les centres des clusters, tout en tenant compte des degrés d'appartenance. Initialement, K centres de clusters sont sélectionnés aléatoirement. Pour chaque point de données, des degrés d'appartenance à chaque cluster en fonction de leur distance sont calculés. Les centres de clusters sont recalculés en tenant compte des degrés d'appartenance, en utilisant la moyenne pondérée des points, où les poids sont les degrés d'appartenance. Les étapes de calcul des degrés d'appartenance et de mise à jour des

centres se répètent jusqu'à ce que les centres de clusters se stabilisent ou que le changement soit inférieur à un seuil prédéfini.

2.7.2 Logique Floue

Logique Floue I et II La logique floue [66], introduite par Lotfi Zadeh dans les années 1960, est une extension de la logique classique qui permet de modéliser l'incertitude et l'imprécision. Cette approche est particulièrement utile dans des situations où les catégories ne sont pas clairement définies, permettant ainsi une meilleure représentation des connaissances humaines. Les systèmes basés sur la logique floue sont composés de règles floues qui relient les entrées à des sorties. Ces règles permettent de gérer des situations complexes en fournissant des décisions nuancées qui reflètent les variations des paramètres d'entrée.

Le principe de fonctionnement d'un système de logique floue repose sur trois étapes principales : la fuzzification, l'application des règles floues et la défuzzification. Dans la fuzzification, les valeurs d'entrée crisp (précises) sont converties en valeurs floues. Cela se fait à l'aide de fonctions d'appartenance qui définissent à quel degré une valeur appartient à une catégorie donnée. Les règles floues sont ensuite appliquées pour établir des relations entre les entrées et les sorties. Ces règles, souvent sous forme de déclarations "Si... Alors...", permettent au système de traiter des valeurs floues et de générer des résultats intermédiaires en fonction des degrés d'appartenance. Enfin, les résultats flous sont convertis en une valeur crisp grâce à la défuzzification.

La logique floue de type II est une extension de la logique floue classique (type I) qui permet de gérer non seulement l'incertitude des données, mais aussi l'incertitude liée à l'appartenance des valeurs elles-mêmes. Cela se fait par l'introduction d'une dimension supplémentaire dans les ensembles flous, offrant ainsi une modélisation plus flexible des situations complexes.

2.7.3 Méthode de Réduction de la Dimensionnalité

Carte auto-organisatrice de Kohonen La carte auto-organisatrice de Kohonen (Self-Organizing Map, ou SOM) [67] est un algorithme principalement utilisé pour la réduction de dimensions et la visualisation de données complexes. Ce modèle cartographique projette des données de haute dimension dans un espace de dimension inférieure, souvent bidimensionnel, tout en conservant la structure topologique des

données. Cela en fait un outil puissant pour la visualisation de clusters et l'analyse exploratoire de données.

Une SOM est composée d'un réseau de neurones disposés en grille. Chaque neurone est associé à un vecteur de poids de même dimension que les données d'entrée. Les poids des neurones sont initialisés, généralement avec des valeurs aléatoires ou selon une distribution proche des données d'entrée. À chaque étape de l'apprentissage, une donnée d'entrée est sélectionnée. Le neurone dont le vecteur de poids est le plus proche de la donnée d'entrée est désigné comme le neurone gagnant (Best Matching Unit, BMU), selon une métrique de distance. Les poids du BMU et des neurones environnants sont ajustés pour les rapprocher de la donnée d'entrée. Cette mise à jour se fait de manière plus importante pour le BMU et diminue progressivement avec la distance dans la grille. Les étapes précédentes se répètent pour chaque donnée, avec un taux d'apprentissage et une taille du voisinage qui diminuent au fil du temps, jusqu'à ce que la carte converge.

2.7.4 Ensemble Learning

Random Forest L'algorithme Random Forest [68] est largement utilisé pour des tâches de classification et de régression. Il repose sur l'idée de créer une forêt d'arbres de décision, chaque arbre étant construit à partir d'un échantillon aléatoire des données.

Le Random Forest fonctionne selon plusieurs étapes clés, à savoir : l'Échantillonnage, la Construction des Arbres, et enfin la Prédiction. À partir de l'ensemble de données d'entraînement, plusieurs échantillons aléatoires (appelés "bootstrap samples") sont générés. Chaque échantillon est utilisé pour créer un arbre de décision distinct. Ensuite, chaque arbre de décision est construit en utilisant une méthode de séparation (ou "splitting") qui choisit parmi un sous-ensemble aléatoire de caractéristiques. Cela introduit une diversité parmi les arbres et améliore la robustesse du modèle. Enfin, pour effectuer des prédictions, chaque arbre de la forêt donne une "vote" pour la classe (dans le cas de la classification) ou prédit une valeur (dans le cas de la régression). La classe finale (ou la valeur prédite) est déterminée par la majorité des votes (pour la classification) ou la moyenne des prédictions (pour la régression).

2.7.5 Neuro-Fuzzy

NFIS L'algorithme Système d'Inférence Neuro-Flou (Neuro-Fuzzy Inference System, NFIS) [69] est une combinaison des réseaux de neurones et des systèmes flous pour créer des modèles d'inférence capables de traiter l'incertitude et la complexité des systèmes. Il est utilisé pour des tâches telles que la classification, la régression et la prise de décision, notamment dans des environnements où les données sont incertaines ou imprécises. Les systèmes flous peuvent être fusionnés avec n'importe quel type de réseau neuronal (par exemple, CNN ou RNN).

2.7.6 Algorithmes d'apprentissage par renforcement traditionnels

Q-learning Le Q-learning [70] est un algorithme qui permet à un agent d'apprendre à prendre des décisions optimales en interagissant avec un environnement. L'agent apprend à maximiser la récompense cumulée en ajustant une fonction de valeur, appelée fonction Q, qui évalue la qualité des actions dans différents états.

L'algorithme suit un processus itératif où l'agent choisit une action à partir de l'état actuel. Après avoir exécuté l'action, il reçoit une récompense, et une fonction Q est mise à jour en fonction de la récompense et des valeurs maximales des actions possibles dans l'état suivant.

2.7.7 Systèmes Neuro-Flous Renforcés

RL-FIS Le RL-FIS (Reinforcement Learning Fuzzy Inference System) [71] est une approche hybride qui combine les principes de l'apprentissage par renforcement (RL) avec les systèmes d'inférence floue (FIS). Cette combinaison permet de bénéficier à la fois de la capacité d'apprentissage adaptatif de l'algorithme d'apprentissage par renforcement et de la gestion de l'incertitude et de l'imprécision des systèmes flous. Le RL-FIS est particulièrement utile dans des environnements complexes et dynamiques où les décisions doivent être prises en tenant compte de l'imprécision et des ambiguïtés.

Le fonctionnement du RL-FIS repose sur plusieurs étapes, à savoir : la Modélisation Floue, l'Apprentissage par Renforcement, et la Mise à Jour des Règles Floues. Initialement, un système d'inférence floue est utilisé pour modéliser la relation entre les états et les actions. Les entrées sont floues et sont définies par

des ensembles flous, ce qui permet de capturer l'incertitude des données. Ensuite, un agent apprend à maximiser une récompense cumulative en interagissant avec l'environnement. Le RL-FIS ajuste les règles floues et les paramètres du système d'inférence floue en fonction des récompenses obtenues suite aux actions entreprises. Enfin, le processus d'apprentissage permet d'optimiser les règles floues en adaptant les membres de l'ensemble et les poids associés aux règles, ce qui améliore la précision du système dans la prise de décision.

2.7.8 Algorithmes d'apprentissage par Renforcement Profond

DQN Le DQN (Deep Q-Network) [72] est une méthode hybride qui combine l'apprentissage par renforcement traditionnel, notamment le Q-learning, avec les techniques de l'apprentissage profond.

Le DQN repose sur plusieurs concepts clés, à savoir : une Fonction Q Approximée, la Mémorisation de l'Expérience, un Échantillonnage par Minibatch, et une Cible Fixe. Au lieu de maintenir une table Q pour représenter les valeurs d'action pour chaque état, le DQN utilise un réseau de neurones pour approximer la fonction Q. Les entrées du réseau sont les états de l'environnement, et les sorties correspondent aux valeurs Q pour chaque action possible. Un composant essentiel du DQN est la mémoire d'expérience, qui stocke les transitions (état, action, récompense, état suivant). Cette mémoire est utilisée pour l'entraînement du réseau de neurones, permettant de briser la corrélation entre les échantillons d'expérience. Le DQN utilise un échantillonnage par minibatch à partir de la mémoire d'expérience pour entraîner le réseau. Cela permet d'améliorer la stabilité de l'apprentissage. Pour stabiliser l'apprentissage, le DQN emploie un réseau de cible fixe, qui est mis à jour périodiquement avec les poids du réseau principal. Cela réduit les oscillations et les divergences lors de l'entraînement.

D3QN Le D3QN (Double Deep Q-Network) [73] est une extension du DQN qui vise à résoudre le problème de la surévaluation des valeurs d'action.

Contrairement au DQN traditionnel, qui utilise une seule fonction Q pour sélectionner les actions et estimer les valeurs, le D3QN emploie deux réseaux de neurones distincts. L'un est utilisé pour choisir les actions (réseau d'action) et l'autre pour évaluer ces actions (réseau de valeur). Cela réduit le biais de surévaluation qui peut survenir lorsque la même fonction est utilisée pour les deux. Comme le DQN, le D3QN utilise une mémoire d'expérience pour stocker les transitions et

un échantillonnage par minibatch pour l'entraînement, garantissant que les échantillons sont diversifiés et brisent les corrélations. Le D3QN utilise également un réseau de cible fixe, qui est mis à jour périodiquement, permettant de stabiliser le processus d'apprentissage en évitant les oscillations fréquentes.

AC-DRL L'AC-DRL (Actor-Critic Deep Reinforcement Learning) [74] est une méthode d'apprentissage par renforcement qui combine les approches des modèles basés sur des valeurs (comme Q-learning) et des modèles basés sur des politiques (comme les méthodes de politique directe). Ce cadre hybride vise à tirer parti des avantages des deux paradigmes pour améliorer l'efficacité de l'apprentissage et la qualité des politiques apprises.

Le cadre AC-DRL repose sur deux composants principaux : l'acteur (actor) et le critique (critic). L'acteur est responsable de la sélection des actions en fonction de l'état actuel de l'environnement. Il apprend une politique (une fonction qui mappe les états aux actions) qui maximise la récompense cumulée. Le critique évalue les actions prises par l'acteur en estimant la valeur des états (ou la valeur des actions) en utilisant une fonction de valeur. Il fournit des retours à l'acteur sur la qualité des actions choisies, ce qui lui permet d'affiner la politique, et met à jour sa fonction de valeur en fonction des récompenses reçues et des estimations de l'acteur.

2.8 Applications du ML

Les applications du ML sont vastes et en constante évolution, impactant presque tous les secteurs. La capacité de ces algorithmes à traiter des volumes massifs de données et à en extraire des informations pertinentes ouvre la voie à des innovations qui peuvent transformer notre quotidien.

Santé Le ML joue un rôle essentiel dans le secteur de la santé, où il est utilisé pour prédire des maladies, analyser des images médicales et personnaliser les traitements. Les algorithmes peuvent aider à identifier des modèles dans les données des patients, améliorant ainsi les diagnostics [75].

Marketing Le ML transforme le marketing en permettant une personnalisation accrue des campagnes publicitaires. En analysant les données des clients, les algorithmes peuvent segmenter les audiences, prévoir les comportements d'achat et optimiser les recommandations de produits, augmentant ainsi l'engagement et les conversions [76].

Finance Dans le secteur financier, le ML est utilisé pour la détection des anomalies, l'évaluation des risques et la gestion des portefeuilles d'investissement. Les modèles de ML analysent les comportements des transactions pour repérer les fraudes et aider à prédire les tendances comportementales des clients [77].

Agriculture Le ML est également appliqué en agriculture, où il aide à optimiser les rendements des cultures en analysant les données climatiques, du sol et des rendements passés. Les drones et les capteurs équipés d'algorithmes de ML permettent une surveillance précise des cultures [78].

Robotique Dans le domaine de la robotique, le ML permet aux robots d'apprendre de leurs expériences et d'interagir avec leur environnement de manière autonome. Des algorithmes d'apprentissage par renforcement sont utilisés pour développer des robots capables de naviguer, de manipuler des objets et de prendre des décisions en temps réel [79].

2.9 Conclusion

Le machine learning se révèle être une technologie puissante avec un large éventail d'applications. Les algorithmes de ML, qu'ils soient supervisés, non supervisés ou basés sur le renforcement, offrent des solutions robustes qui touchent à des secteurs variés tels que la santé, la finance, le marketing, l'agriculture et la robotique.

Dans ce chapitre, nous avons exploré le rôle central du clustering en ML, une technique clé permettant d'identifier des motifs cachés dans des ensembles de données complexes et de regrouper des éléments en fonction de leur similarité. Cette approche s'avère particulièrement utile dans les environnements où une structure sous-jacente doit être découverte, comme c'est souvent le cas dans les applications de grande envergure telles que l'IoT.

Dans le prochain chapitre, nous approfondirons l'utilisation du clustering basé sur le ML dans le contexte de l'IoT. Nous explorerons comment ces techniques sont employées pour améliorer la gestion et l'optimisation des réseaux dans un environnement de plus en plus connecté et intelligent.

Chapitre 3

Protocoles de clustering basés ML dans l'IoT : Etat de l'art

3.1 Introduction

L'IoT est devenu un domaine en plein essor, avec une prolifération d'appareils interconnectés générant et transmettant de vastes volumes de données. Cette expansion pose des défis importants en matière de gestion de ces réseaux complexes, où la communication, le traitement et la prise de décision doivent être réalisés de manière efficace. Afin d'optimiser ces réseaux et d'assurer une gestion intelligente des ressources, l'approche de clustering a gagné en popularité. Le clustering permet de diviser le réseau en plusieurs sous-ensembles de dispositifs ou clusters, facilitant ainsi la gestion des données et l'allocation des ressources, tout en améliorant l'efficacité énergétique.

Parmi les nombreuses méthodes de clustering, celles basées sur le ML se sont particulièrement distinguées ces dernières années [3]. Le ML, grâce à sa capacité à analyser et à apprendre à partir de données complexes, offre des solutions prometteuses pour la gestion et l'organisation des réseaux de l'IoT. En particulier, les techniques de clustering pilotées par des algorithmes d'apprentissage automatique permettent de prendre des décisions plus précises et mieux adaptées aux conditions changeantes du réseau.

La figure 3.1 illustre la tendance croissante du nombre de publications académiques qui exploitent le ML pour le clustering dans l'IoT. Cette progression témoigne de l'intérêt grandissant de la communauté scientifique pour cette approche, en raison de ses nombreux avantages par rapport aux méthodes classiques.

Ce chapitre présente les principaux protocoles de clustering basés sur le ML dans les réseaux IoT, accompagnés d'une taxonomie des algorithmes correspondants. Nous

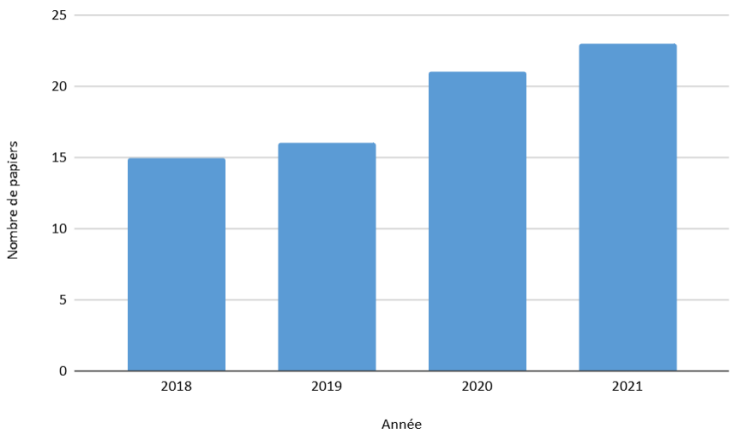


FIGURE 3.1 – Investigation du clustering basé sur le ML dans l'IoT de 2018 à 2021.

proposons également une analyse statistique de l'intégration des techniques de ML pour le clustering dans différents systèmes IoT, en soulignant les questions ouvertes.

3.2 Taxonomie des protocoles de clustering basés ML dans l'IoT

Un certain nombre d'approches basées sur le ML pour le clustering dans l'IoT existent dans la littérature. Dans notre étude, nous avons regroupé ces approches selon l'approche d'apprentissage utilisée : supervisées, non supervisées et par renforcement. Chaque catégorie inclut différentes familles d'algorithmes, comme illustré dans la figure 3.2.

3.3 Revue de la littérature

Dans cette section, nous explorons les différents protocoles de clustering dans les réseaux IoT, classés selon l'approche d'apprentissage utilisée : non supervisée, supervisée et par renforcement. Chaque catégorie regroupe différentes familles d'algorithmes, offrant une vue structurée des méthodes employées.

3.3.1 Approches non supervisées

Les protocoles de clustering basés sur une approche non supervisée utilisent des techniques qui ne nécessitent pas de données étiquetées pour le processus de formation des clusters. Ces protocoles exploitent des algorithmes qui identifient des structures

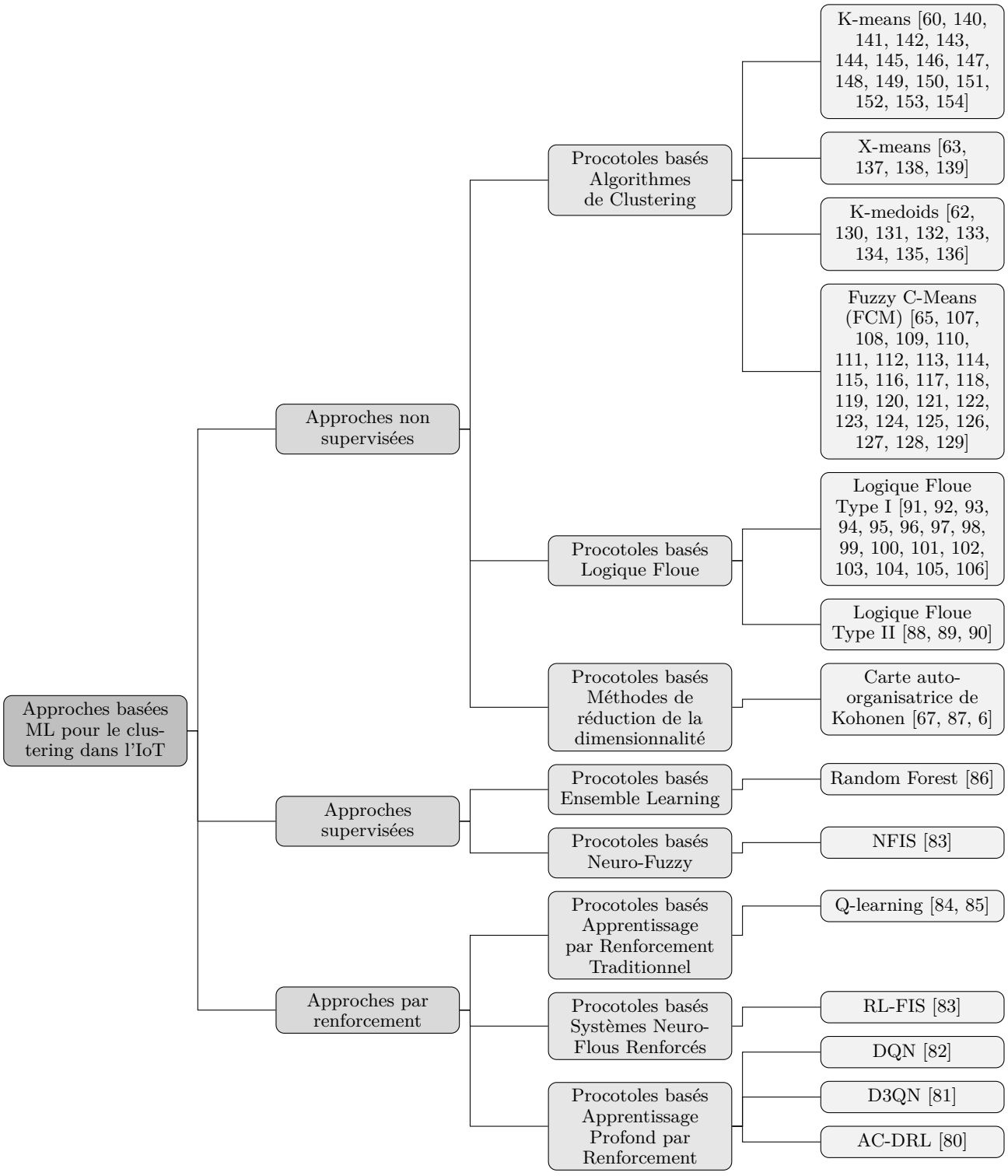


FIGURE 3.2 – Taxonomie des protocoles basés ML pour le clustering dans l'IoT.

TABLE 3.1 – Revue des algorithmes basés sur des approches non supervisées utilisés pour le clustering dans l'IoT.

Méthode	Avantages	Inconvénients
Algorithmes de clustering	- Simplicité de mise en œuvre, même pour des volumes de données considérables. - Efficaces pour des données non étiquetées, permettant de diviser les données en clusters distincts. - Capacité à révéler des structures de données sous-jacentes sans connaissances préalables.	- Sensibilité à l'initialisation : le choix du nombre de clusters est souvent nécessaire à l'avance et peut biaiser les résultats. - Sensibilité élevée aux outliers (données aberrantes), ce qui peut détériorer la qualité du clustering. - Certains algorithmes, comme K-means, peuvent ne pas bien gérer des clusters de formes irrégulières ou de tailles différentes.
Logique floue	- Capacité à gérer l'incertitude et l'ambiguïté dans les données, offrant une grande flexibilité. - Peut modéliser des systèmes complexes où les frontières entre les classes ne sont pas nettes. - Approche robuste pour des environnements non déterministes ou des données incomplètes.	- Complexité accrue dans la conception et l'ajustement des règles floues, notamment pour des systèmes avec un grand nombre de variables. - Sensibilité aux paramètres d'appartenance, ce qui peut rendre l'optimisation difficile. - Peut exiger des temps de calcul plus longs, particulièrement pour de grandes bases de données.
Méthodes de réduction de la dimensionnalité	- Permettent une visualisation intuitive des données en haute dimension en les réduisant à deux ou trois dimensions. - Capables de regrouper des données similaires tout en préservant la topologie des données originales. - Utiles pour la détection de motifs ou de clusters dans des ensembles de données complexes.	- Performances limitées pour des ensembles de données très volumineux ou complexes, en particulier pour des formes non linéaires. - Nécessitent des paramètres ajustés avec soin, tels que la taille de la carte ou les poids d'interconnexion pour les cartes de Kohonen, pour de bonnes performances. - Peu efficaces pour des données qui n'ont pas de relations spatiales évidentes.

cachées ou des motifs sous-jacents dans les données, ce qui est particulièrement utile pour les réseaux IoT, où les données étiquetées peuvent être rares ou inexistantes. L'un des avantages majeurs de ces approches est leur capacité à gérer de grands ensembles de données hétérogènes avec précision. Comme le montre la figure 3.2, parmi les familles d'algorithmes utilisées, nous distinguons les algorithmes de clustering, la logique floue et les méthodes de réduction de la dimensionnalité (voir la section 2.7, page 30). Le tableau 3.1 passe en revue ces catégories et met en évidence leurs avantages et inconvénients.

Protocoles basés algorithmes de clustering

De nombreuses études ont exploité les algorithmes de clustering pour améliorer le routage dans l'IoT.

L'algorithme K-means a été exploré pour optimiser la sélection des CHs et la for-

mation des clusters dans les réseaux IoT. Khandelwal et Jain [60] ont exploité K-means pour la sélection des CHs dans leur approche de sélection et de classement appropriés des CHs pour une communication efficace avec la station de base. Jain *et al.* [140] ont développé un algorithme de clustering basé sur K-means visant à améliorer l'efficacité énergétique en re-sélectionnant les CHs parmi les plus viables, selon leur proximité avec la station de base, augmentant ainsi le débit et réduisant la latence. Bhushan *et al.* [144] ont intégré K-means dans leur approche de sélection des CHs basée sur un algorithme génétique, où la population du GA est initialisée par K-means pour une évaluation optimisée des CHs. Razzaq *et al.* [145] ont utilisé K-means pour la formation de clusters dans leur protocole de routage hiérarchique pour les WSNs basé sur la taille optimale des paquets. De même, Pavithra et Babu [146] ont utilisé K-means dans leur protocole de routage hiérarchique. Face aux défis de la fiabilité du canal radio et de la dispersion des nœuds, El Khediri *et al.* [149] ont proposé l'intégration de K-means pour structurer leur réseau en clusters dans leur protocole d'amélioration de la localisation des nœuds pour les WSNs. De même, Jain et Kumar [152] ont introduit K-means pour la formation des clusters dans leur protocole dynamique de routage hiérarchique. Omeke *et al.* [153] ont également utilisé K-means pour la formation des clusters dans leur protocole de routage hiérarchique conçu pour maximiser la durée de vie des WSNs sous-marins. Srivastava *et al.* [150] ont, quant à eux, adopté une approche hybride combinant K-means et la recherche gloutonne (Greedy Best-First Search, GBFS) [155] pour la formation des clusters dans leur protocole de routage économe en énergie et à contrôle de congestion. Rezaeipanah *et al.* [154] ont combiné K-means avec l'algorithme du modèle de développement de source libre (Open Source Development Model Algorithm, ODMA) [156] pour une formation dynamique de clusters dans leur protocole de routage hiérarchique multi-sauts. Dhand et Tyagi [147] ont intégré K-means pour la formation des clusters dans leur protocole de routage sécurisé pour les WSNs hiérarchiques. Lehsaini et Benmahdi [141] ont modifié K-means en ajoutant une phase de réaffiliation, garantissant des clusters équilibrés dans leur protocole de routage hiérarchique pour les WSNs. Pour une flexibilité accrue, Zhu *et al.* [151] ont développé une version soft-K-means, où un nœud peut appartenir à plusieurs clusters avec des degrés d'appartenance variables. Une réaffectation des nœuds membres est réalisée afin d'équilibrer le nombre de nœuds par cluster. De plus, ils ont intégré les algorithmes de clustering par recherche rapide et identification des pics de densité (Clustering by Fast Search and Find of Density Peaks, CFSFDP) [157] et de l'estimation de la densité par noyau (Kernel Density Estimation, KDE) [158] pour améliorer la sélection des centroïdes. Srikanth *et al.* [142] ont inté-

gré K-means dans leur protocole de clustering pour les WSNs, remplaçant la sélection aléatoire des centroïdes par la méthode du point médian, visant à optimiser la sélection initiale des centroïdes. Kaur *et al.* [143] ont, quant à eux, proposé une méthode de point médian modifiée, permettant de trouver les centroïdes initiaux en considérant à la fois les dimensions horizontales et verticales. Dong *et al.* [148] ont, quant à eux, exploité K-means pour effectuer un clustering inégal dans leur protocole de routage conçu pour les WSNs dans les mines de charbon.

L'algorithme X-means a été employé également pour améliorer la formation des clusters dans les réseaux IoT. Dans l'étude de Radwan *et al.* [63], les auteurs emploient X-means pour le clustering du réseau. Ils appliquent d'abord l'algorithme K-means pour diviser le réseau en deux clusters, puis subdivisent chaque centroïde en plusieurs positions. Cette méthode évalue le processus de clustering afin de déterminer si une nouvelle subdivision est nécessaire pour identifier le positionnement et le nombre optimal de clusters. Une plage est utilisée pour limiter le nombre minimal et maximal de clusters. Dans des travaux ultérieurs [138, 137], une plage dynamique pour le nombre de clusters a été adoptée, permettant de surmonter les contraintes liées à la plage fixe. Plus récemment, Radwan *et al.* [139] ont introduit une fonction de fitness pour guider les décisions de division des clusters dans l'algorithme X-means.

L'algorithme K-medoids a de même été utilisé pour améliorer la sélection des CHs et la formation des clusters dans les réseaux IoT. Sharma *et al.* [62] ont intégré l'algorithme de clustering K-medoids pour la formation des clusters et la sélection des CHs dans leur protocole de clustering pour les WSNs, où le nœud médianoïde a été choisi comme CH. Gupta *et al.* [130] ont intégré l'algorithme K-medoids pour la formation des clusters parmi lesquels les CHs sont sélectionnés par la suite. Elhosny et Shankar [131] ont intégré l'algorithme K-medoids pour le clustering dans les réseaux IoV. De la même manière, Chen et Hengjinda [132] ont intégré l'algorithme de clustering K-medoids pour la formation de clusters afin d'améliorer l'efficacité de la communication V2V. Iswarya et Radha [134] ont intégré l'algorithme de clustering K-medoids pour la formation de clusters afin d'optimiser les communications V2V. Shanmugam et Kaliaperumal [133] ont intégré K-medoids dans leur protocole de routage opportuniste conçu pour les WSNs. Dans [136], les auteurs ont intégré l'algorithme de clustering K-medoids pour la formation de clusters dans leur protocole de routage pour les WSNs. Les nœuds médianoïdes jouent le rôle de passerelles entre les CHs et la station de base. Dans la même lignée, Faid *et al.* [135] ont incorporé l'algorithme K-medoids pour la formation de clusters dans leur protocole de clustering pour les WSNs.

L'algorithme FCM a été mis en œuvre également pour la sélection efficace des CHs et la formation des clusters dans les réseaux IoT. Chelbi et al. [107] ont intégré l'algorithme FCM pour la formation de clusters dans leur protocole de routage hiérarchique basé sur le contrôle de couverture pour les WSNs. Dans chaque cluster, un nœud collecteur d'énergie est placé dans la grille la plus proche du centroïde. Chouhan et Jain [124] ont, quant à eux, intégré l'algorithme FCM pour la sélection des CHs dans leur protocole de clustering pour les WSNs. Panchal et Singh [125] ont développé un protocole de routage hiérarchique à trois niveaux basé sur FCM. Dans cette architecture, les nœuds normaux transmettent leurs données collectées aux CHs, qui les envoient ensuite aux Grid Heads (GHs), avant que ces derniers ne transmettent les données à la station de base. Une fonction de fitness FCM est utilisée pour sélectionner de manière optimale les GHs et les CHs. Moussaoui et al. [127] ont proposé un algorithme de routage énergétique basé sur des chaînes pour les WSNs sous-marins. La formation des clusters est appuyée par l'algorithme FCM. Cumino et al. [121] ont introduit un modèle de gestion des messages de contrôle pour l'IoFT basé sur des clusters. Les entités de l'IoFT sont divisées en clusters basés sur l'algorithme FCM. Augustine et Ananth [120] ont proposé une approche basée sur un FCM à noyau de Taylor (Taylor KFCM), une version modifiée de la méthode FCM pour la sélection des CHs dans les WSNs. Cette approche sélectionne les CHs en fonction d'un facteur d'acceptabilité prenant en compte le niveau d'énergie, la distance jusqu'à la station de base et le niveau de confiance du nœud. Dans la même thématique, Salman et al. [114] ont intégré KFCM, mais ici pour la formation de clusters dans les WSNs, avec un CH et un CH de secours sélectionnés sur la base d'un système d'inférence floue. Thakurta et Roy [65] ont, quant à eux, intégré l'algorithme FCM pour la formation de clusters dans leur protocole de clustering décentralisé pour les WSNs. De la même manière, Wang et al. [109] ont introduit l'algorithme FCM pour la formation de clusters de manière distribuée dans leur protocole de clustering pour les WSNs. Dans leur étude, Jain et al. [111] ont intégré l'algorithme FCM pour établir un nombre optimal de clusters dans leur protocole de clustering pour les WSNs. Wan et al. [112] ont intégré l'algorithme FCM pour la formation de clusters sur la base de la similarité des données dans leur protocole de clustering pour les WSNs. Mamatha et Aishwarya [113] ont intégré FCM pour la formation de clusters dans leur protocole de routage basé sur clusters conçu pour les réseaux IoV, introduisant un processus de clustering optimisé avec des unités routières collectant les données des CHs. Sharma et al. [116] ont introduit FCM pour la formation de clusters dans leur schéma de clustering pour les WSNs. Rajput et Kumaravelu [117] ont intégré l'algorithme FCM pour la formation des clusters dans leur protocole conçu

pour les WSNs dans l'agriculture. FCM est exécuté de manière distribuée au niveau de chaque nœud. Bensaid et al. [118] ont intégré l'algorithme FCM pour la formation de clusters dans leur protocole pour les WSNs dans les applications IoT. Kumar et al. [119] ont intégré l'algorithme FCM pour partitionner les hubs de capteurs dans leur protocole de clustering pour les WSNs compromis afin de contrôler les risques. Karunkuzhali et al. [129] ont intégré l'algorithme FCM pour la formation de clusters dans leur protocole conçu pour surveiller les conditions environnementales dans le cadre de l'agriculture intelligente avec les WSNs. Dans [128], l'algorithme FCM de densité (DFCM) a été employé pour la formation de clusters, prenant en compte la densité des nœuds lors de la sélection des centroïdes initiaux, dans leur protocole de regroupement pour les WSNs dans les smart grids. Su et Zhao [108] ont intégré l'algorithme FCM pour la formation de clusters dans leur protocole de clustering pour les WSNs. Les centroïdes initiaux sont choisis par l'emplacement vicinal avec des nœuds denses. Mondal et al. [110] ont introduit l'algorithme FCM brut (RFCM) pour diviser le réseau en clusters dans leur protocole de routage basé sur des chaînes pour les WSNs. Srivenkateswaran et Sivakumar [115] ont proposé une approche hybride K-means et FCM pour la formation de clusters dans leur protocole d'agrégation sécurisée de données basé sur des clusters dans les WSNs. Une nouvelle approche a été introduite par Hassan et al. [122], améliorant l'algorithme FCM pour construire des clusters équilibrés de nœuds de capteurs. Les centroïdes finaux de l'algorithme FCM sont recalculés de manière à trouver des clusters équilibrés. Reegan et Kabila [126] ont intégré l'algorithme FCM pour la formation de clusters dans leur protocole assurant la transmission sécurisée des données tout en réduisant le temps de transmission dans les WSNs. Enfin, Chandrawanshi et al. [123] ont intégré l'algorithme FCM++ pour la formation de clusters dans leur protocole de routage pour les WSNs, où les centroïdes sont initialisés via K-means++ [159].

Protocoles basés Logique Floue

De nombreuses études ont exploité la logique floue pour améliorer le routage dans l'IoT.

La logique floue de type I a été explorée pour optimiser la sélection des CHs. Wang *et al.* [91] ont proposé un protocole de clustering économe en énergie basé sur la logique floue pour l'IoT. Dans ce protocole, les CHs déterminent le rayon de communication en se basant sur des entrées floues, telles que la densité des nœuds et la distance à la station de base. La probabilité d'être CH est calculée localement en prenant en compte l'énergie actuelle et la distance totale. Rahim et Chrysostomou [92] ont introduit un protocole

de clustering à charge équilibrée basé sur la logique floue. L'inférence floue est employée pour la sélection des CHs, basée sur la distance par rapport à la station de base, l'énergie résiduelle et la distance moyenne par rapport aux voisins. Pour améliorer la consommation d'énergie dans l'IoT, Mazinani *et al.* [93] ont intégré la logique floue pour le routage. Ils ont considéré l'énergie résiduelle, le nombre de nœuds et la distance de chaque nœud comme des critères flous pour choisir les CHs. Mirzaie et Mazinani [94] ont proposé un protocole de clustering économe en énergie basé sur la logique floue pour l'IoT, intégrant des nœuds hétérogènes. Les critères tels que l'énergie résiduelle, le nombre de voisins et la distance à la station de base sont utilisés pour sélectionner les CHs les plus adaptés, tout en adoptant un seuil fixe pour éviter une consommation d'énergie déséquilibrée. Hamzah *et al.* [95] ont présenté un système basé sur la logique floue pour la sélection des CHs, prenant en compte des facteurs tels que l'énergie résiduelle, l'adéquation de l'emplacement, la densité, le compactage et la distance à la station de base. Rajput et Babu [96] ont développé un protocole de clustering distribué basé sur la logique floue, visant à améliorer la consommation d'énergie dans l'IoT tout en maximisant la zone de couverture. Ils ont utilisé la logique floue pour sélectionner les CHs en fonction du niveau d'énergie et de la qualité du signal. De plus, la charge des CHs est partagée avec des super-CHs pour éviter leur mort prématurée, ces derniers étant sélectionnés aussi par la logique floue. Les paramètres pris en compte incluent l'énergie et la distance normalisée par rapport à la station de base. Verma *et al.* [98] ont développé un schéma de clustering efficace basé sur la logique floue pour l'IoT utilisant un sink mobile. L'élection des CHs repose sur la probabilité d'énergie moyenne et l'impact de l'énergie des nœuds à chaque rotation, favorisant ainsi une distribution équitable de l'énergie. Un système de logique floue est utilisé pour sélectionner les super-chefs de groupe parmi l'ensemble initial des CHs possibles, en considérant quatre facteurs : la puissance de la batterie du capteur, la mobilité de la station de base, la centralité des CHs et l'énergie moyenne. Ch et Budyal [99] ont proposé une collecte de données économe en énergie basée sur la logique floue dans l'IoT, utilisant des éléments mobiles. Les CHs sont sélectionnés en tenant compte de paramètres linguistiques, tels que la distance à un CH voisin, l'énergie résiduelle et le coût de communication au sein du cluster. Karunanithy et Velusamy [100] ont proposé un protocole de collecte de données économe en énergie basé sur des clusters pour l'IIoT. Ce protocole utilise la logique floue pour sélectionner les CHs après avoir déterminé leur nombre optimal à partir des informations collectées localement parmi un ensemble de CHs candidats. La logique floue est ensuite employée pour évaluer les probabilités des CHs finaux en tenant compte de trois critères : la distance moyenne par rapport aux

nœuds membres, l'énergie résiduelle et le nombre de nœuds voisins. Mishra *et al.* [101] ont conçu un protocole de clustering économe en énergie basé sur la logique floue pour réaliser un clustering inégal dans l'IoT. Le processus de clustering est exécuté différemment en utilisant un système d'inférence floue, avec trois types de clustering distincts qui se succèdent tout au long des rounds. Dans le premier type, l'énergie résiduelle et le degré des nœuds sont pris en compte. Le deuxième type repose sur la distance par rapport à la station de base ainsi que sur l'énergie résiduelle. Enfin, le troisième type considère l'énergie et la distance minimale des CHs. Dans [102], une optimisation du protocole HEED intègre la logique floue. Le système de logique floue sélectionne les CHs les plus pertinents selon la distance à la station de base, les niveaux d'énergie actuels et la densité des nœuds. Bhushan et Sahoo [103] ont proposé un protocole de clustering économe en énergie basé sur la logique floue pour l'IoT, utilisant des règles floues qui intègrent des paramètres tels que l'énergie résiduelle, la distance moyenne intra-cluster, le degré de compactage, la probabilité d'abandon des paquets et l'historique des nœuds pour une sélection optimale des CHs. Les auteurs de [105] ont introduit un algorithme de clustering intégrant la logique floue. Un système d'inférence déductive basé sur la logique floue a été mis en œuvre pour sélectionner les CHs appropriés en utilisant la force d'attraction, l'énergie résiduelle et la distance. Adnan *et al.* [106] ont développé un protocole de routage multi-sauts en clusters inégaux pour l'IoT, fondé sur la logique floue. Cette méthode crée des clusters inégaux en sélectionnant les CHs à l'aide d'un système flou se basant sur la distance du nœud à la station de base, le nombre de nœuds adjacents à proximité du CH et l'énergie résiduelle. Dwivedi et Sharma [97] ont proposé un protocole de clustering économe en énergie basé sur la logique floue. Deux différentes valeurs d'aptitude floue sont utilisées pour sélectionner les CHs et former les clusters. La sélection des CHs se base sur des facteurs primaires, tels que la distance à la station de base, le nombre de voisins, l'énergie résiduelle et le coût de communication. Dans le processus de formation des clusters, les nœuds qui ne sont pas des CHs utilisent des facteurs secondaires, tels que le nombre de membres et la distance au CH, pour assurer une sélection adéquate de leurs CHs. Enfin, Rajeswari *et al.* [104] ont introduit un nouvel algorithme de routage hiérarchique pour l'IoT intégrant la logique floue. La logique floue est adoptée pour sélectionner les CHs et former les clusters, en utilisant des critères tels que l'énergie actuelle du nœud, la densité du cluster et la distance à la station de base.

La logique floue de type II a été également explorée pour optimiser la sélection des CHs. Alkhliwi et al. [88] ont proposé un protocole de routage hiérarchique intégrant la logique floue de type II. Les règles de logique floue ont été appliquées pour une sélection

efficace des CHs, en prenant en compte le niveau d'énergie, la distance par rapport à la station de base et la densité des dispositifs. Adnan et al. [89] ont conçu un schéma de clustering inégal afin d'optimiser la consommation d'énergie et d'améliorer l'évolutivité du réseau. La stratégie de sélection des CHs est basée sur la logique floue de type II, intégrant des critères tels que la distance à la station de base et l'énergie résiduelle. En outre, la couverture du cluster est déterminée en fonction des entrées de la logique floue afin d'équilibrer la charge des CHs. Sennan et al. [90] ont proposé un algorithme de clustering visant à prolonger la durée de vie de l'IoT en intégrant la logique floue de type II. Cette dernière est appliquée aux paramètres du réseau, à savoir le niveau d'énergie et la distance entre les capteurs et la station de base, afin de garantir la sélection des meilleurs CHs.

Protocoles basés Méthodes de réduction de la dimensionnalité

Les cartes auto-organisatrices de Kohonen ont été adoptées pour la formation de clusters dans les réseaux IoT. Nayak et al. [87] ont utilisé ces cartes pour diviser les nœuds du réseau en clusters. La taille des clusters a ensuite été ajustée à l'aide de l'algorithme K-means afin de déterminer le nombre optimal de clusters, ce qui impacte directement la charge du réseau et l'équilibre de la consommation énergétique. Dans le même ordre d'idées, Merah et al. [6] ont intégré l'algorithme SOM pour la formation de clusters dans leur protocole de routage pour les réseaux IoT dynamiques. Par ailleurs, Merah et al. [67] ont conçu un protocole de clustering écoénergétique basé sur l'algorithme SOM pour les réseaux IoT. Les dispositifs du réseau sont divisés en clusters à l'aide de cet algorithme, et les CHs sont sélectionnés en fonction de leur niveau d'énergie et de leur distance par rapport au neurone gagnant.

3.3.2 Approches supervisées

Les protocoles de clustering basés sur une approche supervisée reposent sur des algorithmes qui nécessitent un ensemble de données étiquetées pour entraîner le modèle. Ces protocoles apprennent à partir d'exemples préalablement étiquetés pour faire des prédictions ou des classifications sur de nouvelles données. Bien que le clustering soit principalement une tâche non supervisée, les approches supervisées peuvent ajouter de la valeur en exploitant des données étiquetées pour mieux former ou affiner les clusters. Comme le montre la figure 3.2, parmi les familles d'algorithmes utilisées, nous distin-

guons les algorithmes d'ensembles et les systèmes neuro-flous. Le tableau 3.2 passe en revue ces catégories et met en évidence leurs avantages et leurs inconvénients.

TABLE 3.2 – Revue des algorithmes basés approches supervisées utilisés pour le clustering dans l'IoT.

Méthode	Avantages	Inconvénients
Ensemble Learning	- Combine plusieurs modèles pour réduire l'erreur, améliorant ainsi la précision globale du modèle. - Robuste face aux données bruitées ou corrompues. - Moins sensible au sur-apprentissage (overfitting) que d'autres algorithmes supervisés. - Adapté à la fois à la classification et à la régression.	- Complexité en termes de calcul, en particulier pour les grands ensembles de données. - Difficile à interpréter, car il ne fournit pas de modèle unique simple. - Peut nécessiter plus de mémoire et de temps pour l'entraînement comparé à d'autres algorithmes plus simples.
Systèmes Neuro-Flous	- Combine les avantages des réseaux de neurones et de la logique floue. - Bien adapté aux environnements dynamiques et incertains. - Peut modéliser des relations complexes entre les variables d'entrée et de sortie.	- Nécessite une optimisation intensive pour ajuster correctement les paramètres et les règles floues. - Temps d'entraînement relativement long en raison de la complexité du modèle. - Peut être plus difficile à mettre en œuvre et à interpréter que d'autres méthodes classiques.

Protocoles basés Ensemble Learning

Les méthodes d'ensemble ont été éxploité pour améliorer le routage dans l'IoT. Rhesa *et al.* [86] ont proposé une approche d'acheminement des données dans les réseaux sans fil intégrant l'algorithme Random Forest. Cette méthode commence par évaluer le niveau d'énergie de chaque nœud afin de classer les à l'aide de l'algorithme Random Forest. Cet algorithme utilise une technique de bagging pour optimiser les performances de classification. Grâce à cette approche, les capteurs sont classés en nœuds primaires et secondaires, selon leur niveau d'énergie respectif, les nœuds primaires ayant une énergie inférieure et les nœuds secondaires une énergie supérieure. Une fois que les nœuds primaires voisins ont collecté les données, les nœuds secondaires sont chargés de les relayer vers la station de base.

Protocoles basés Systèmes Neuro-Flous

Les systèmes neuro-flous ont été exploité pour améliorer le routage dans l'IoT. Thangaramya *et al.* [83] un nouveau protocole de regroupement et de routage basé sur des règles neuro-floues. Les clusters sont formés en tenant compte de quatre paramètres liés

aux CHs, notamment l'énergie résiduelle, la distance avec les nœuds, la distance avec la station de base et le nombre de membre déjà affecté. Ils ont aussi intégrer un CNN avec les règles floues pour un routage optimisé, ou l'apprentissage est effectué sur les paramètres et les règles floues sont utilisées pour l'ajustement des poids.

3.3.3 Approches par renforcement

Les protocoles de clustering basés sur une approche par renforcement sont conçus pour apprendre à prendre des décisions optimales grâce à l'interaction avec l'environnement. Contrairement aux approches supervisées ou non supervisées, une approche par renforcement utilise des récompenses ou des pénalités pour renforcer le comportement souhaité. Dans les réseaux IoT, ces protocoles peuvent adapter dynamiquement les stratégies de clustering en réponse aux changements dans l'environnement. Comme le montre la figure 3.2, parmi les familles d'algorithmes utilisées, nous distinguons l'Apprentissage par Renforcement Traditionnel, les Systèmes Neuro-Flous Renforcés et l'Apprentissage par Renforcement profond (voir page 30). Le tableau 3.3 passe en revue ces catégories et met en évidence leurs avantages et leurs inconvénients.

Protocoles basés apprentissage par Renforcement Traditionnel

Le Q-learning a été utilisé pour optimiser les décisions dans les environnements sans fil. Aliouat *et al.* [85] ont proposé une approche basée sur l'algorithme Q-learning pour trouver dynamiquement un clustering efficace garantissant l'équilibrage de la charge dans un réseau IoT géré par des edge servers. Leur approche repose sur la sélection de trois CHs pour former un ensemble, garantissant trois clusters équilibrés. Les actions consistent à ajuster l'identifiant du CH par incrémentation ou décrémentation, dans le but de maximiser une récompense calculée comme l'inverse de la différence entre le plus grand et le plus petit cluster en termes de densité.

Protocoles basés systèmes Neuro-Flous Renforcés

Les systèmes neuro-flous ont été introduit pour une meilleure gestion des ressources dans les réseaux. Preetha *et al.* [160] a exploité le RL-FIS dans une application de WSNs densément dispersés pour la sélection des nœuds de collecte de données pour chaque cluster, en considérant trois paramètres essentiels, à savoir le chevauchement de voisinage, l'indice de bipartivité et la connectivité algébrique.

TABLE 3.3 – Revue des algorithmes basés approches par renforcement utilisés pour le clustering dans l'IoT.

Méthode	Avantages	Inconvénients
Apprentissage par Renforcement Traditionnel	- Capacité à apprendre à partir d'interactions avec l'environnement sans supervision directe. - Particulièrement adapté aux environnements dynamiques et incertains. - Efficace pour résoudre des problèmes complexes où une solution optimale peut émerger progressivement via essai-erreur.	- Convergence lente, particulièrement pour des problèmes à grande échelle ou complexes. - Nécessite souvent beaucoup d'exploration, ce qui peut être coûteux en termes de temps et de ressources. - Peut devenir instable ou sous-optimal si les récompenses sont mal conçues ou trop différées.
Systèmes Neuro-Flous Renforcés	- Combine les forces de l'apprentissage par renforcement et des systèmes neuro-flous pour une gestion fine de l'incertitude et une meilleure convergence. - Bien adapté aux problèmes où les frontières entre les actions optimales ne sont pas claires. - Capable de traiter des environnements dynamiques et incertains.	- Complexité accrue dans la conception, car il nécessite une combinaison des techniques floues et des réseaux de neurones. - Temps de calcul plus long dû à la nécessité d'ajuster à la fois les paramètres d'apprentissage par renforcement et les paramètres flous. - Difficile à régler et à interpréter, ce qui peut rendre son adoption limitée pour des applications où la simplicité est requise.
Apprentissage par Renforcement Profond	- Capacité à traiter des environnements extrêmement complexes et à haute dimension grâce à l'utilisation de réseaux de neurones profonds. - Apte à apprendre des représentations riches à partir de données brutes sans nécessiter d'ingénierie des caractéristiques. - Succès démontré dans des tâches complexes (jeux vidéo, robotique, systèmes autonomes).	- Exige de grandes quantités de données pour un entraînement efficace. - Coût computationnel très élevé, nécessitant souvent des architectures GPU/TPU. - Instabilité et difficultés à trouver des politiques optimales sans réglage minutieux de l'algorithme et des hyperparamètres.

Protocoles basés apprentissage par Renforcement Profond

L'apprentissage par renforcement profond a été exploité pour prendre des décisions complexes dans les environnements IoT.

Liu *et al.* [82] ont conçu un protocole de clustering dynamique basé sur le DRL pour l'équilibrage de la charge dans les réseaux IoT assistés par des serveurs edge. Un objet se déplace dans la zone du réseau IoT, ce qui incite les appareils IoT voisins à produire des données plus volumineuses que les données normales. Afin d'équilibrer cette charge sur les serveurs un clustering dynamique basé sur DRL est effectué. Le modèle DRL utilisé est le DQN. Pour effectuer le clustering dynamique, deux cluster-cores se déplacent dans le réseau en fonction de cinq actions de mouvement (Up, Down, Left, Right et Stay). Après le déplacement, les dispositif IoT le plus proches des cluster-cores sont désignés

comme les nouveaux CHs et les clusters sont formés sur la base de la proximité.

Dans une extension de leur recherche, Liu *et al.* [81] ont inclus plusieurs objets mobiles dans leur scénario. L'ensemble d'actions possibles a été étendu a sept actions (Up, Down, Left, Right, Stay, Shrink et Expand). Les mouvements des cluster-cores sont guidés par un réseau D3QN basé sur LSTM, entraîné avec des données provenant d'un scénario IoT réel.

Sharif *et al.* [80] ont proposé une approche basée sur AC-DRL pour une sélection efficace des CHs dans l'IoV. La nature dynamique des réseaux IoV pose un défi continu lié à un espace d'états à croissance exponentielle. L'AC-DRL adresse ce problème en permettant aux agents d'estimer simultanément la fonction de valeur état-action via l'acteur et la fonction de critique pour optimiser de manière adaptative la sélection des CHs.

3.4 Bilan

La figure 3.3 présente un diagramme circulaire illustrant la répartition des différentes approches basées sur le ML utilisées dans le clustering.

Comme indiqué, plus de 91 % des approches s'appuient sur des techniques d'apprentissage non supervisé (UL), démontrant ainsi leur prédominance dans ce domaine. En revanche, les approches basées sur l'apprentissage par renforcement (RL) ne représentent qu'un peu plus de 6 % de l'ensemble, tandis que les méthodes supervisées (SL) constituent seulement un peu plus de 2 %. Cette distribution met en évidence la tendance vers des algorithmes non supervisés, souvent privilégiés en raison de leur capacité à traiter des données sans étiquettes et de leur flexibilité dans divers scénarios d'application.

La figure 3.4 présente une analyse des principaux algorithmes de ML utilisés pour le clustering dans les réseaux IoT.

Comme mentionné précédemment, il ressort clairement que l'approche non supervisée est la plus largement adoptée. En particulier, les algorithmes de clustering K-means, K-medoids et FCM ont trouvé une large application. Leur popularité s'explique par leur simplicité mathématique, leur rapidité de convergence et leur facilité d'implémentation. Cependant, il est important de noter que ces méthodes reposent sur des paramètres d'initialisation, tels que le nombre de clusters et les centroïdes initiaux, qui doivent être spécifiés avec précision pour garantir de bons résultats. Les protocoles basés sur la logique floue se démarquent également dans la littérature concernant le clustering dans l'IoT. Cette approche est avantageuse, car elle permet de prendre des décisions en temps

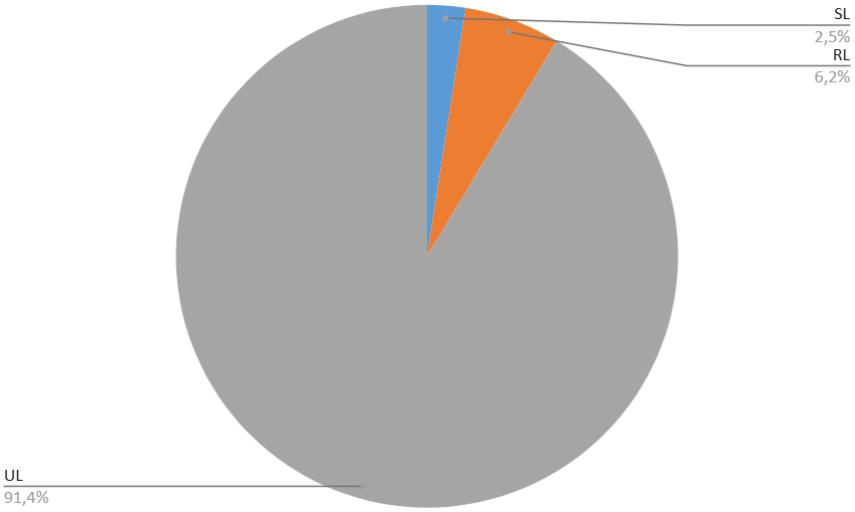


FIGURE 3.3 – Dominance des techniques de ML pour le clustering dans l'IoT.

réel en intégrant divers paramètres. Par exemple, lors du processus de sélection du CH et de l'affectation des nœuds, il est manifeste que la précision s'améliore lorsque plusieurs facteurs sont pris en compte. De plus, l'incertitude inhérente à certaines données est efficacement gérée grâce à l'application de règles floues, ce qui renforce la fiabilité de la prise de décision.

En ce qui concerne l'approche par renforcement, plusieurs études ont exploré cette voie. Parmi elles, le Q-learning et ses variantes sont les plus utilisés. Cette méthode est particulièrement polyvalente, car elle offre la possibilité de fournir des actions de contrôle qui répondent à plusieurs critères, facilitant ainsi la résolution de problèmes d'optimisation multicritères.

Enfin, il convient de souligner que peu de travaux ont recours à l'approche supervisée. En effet, ces méthodes nécessitent des données d'entrée spécifiques.

La figure 3.5 montre la prédominance des catégories d'IoT dans les articles étudiés.

Différents systèmes d'IoT ont été impliqués, à l'exception de l'IoMT et de l'IoNT. Plus précisément, dans l'IoMT, le ML est déjà utilisé dans diverses situations dans le domaine des soins de santé [161, 162, 163]. Toutefois, l'incorporation du clustering basé sur le ML dans ce type de réseau n'a pas encore été mise en œuvre. Dans l'IoNT, la rareté de l'incorporation de techniques de ML reste un défi important en raison de la faisabilité complexe de ces techniques dans ces systèmes extrêmement contraints.

Bien que les techniques de ML aient été largement exploitées pour améliorer les performances des réseaux IoT, leur mise en œuvre peut poser différents problèmes qui

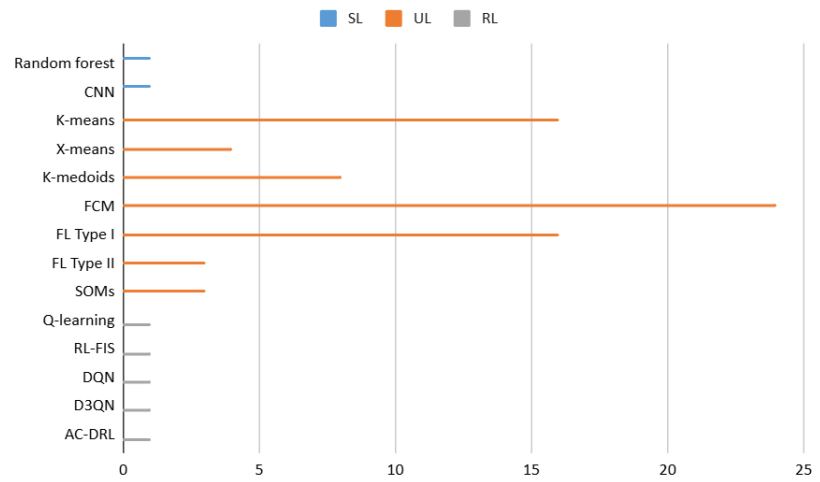


FIGURE 3.4 – Principaux algorithmes de ML utilisés pour le clustering dans l'IoT.

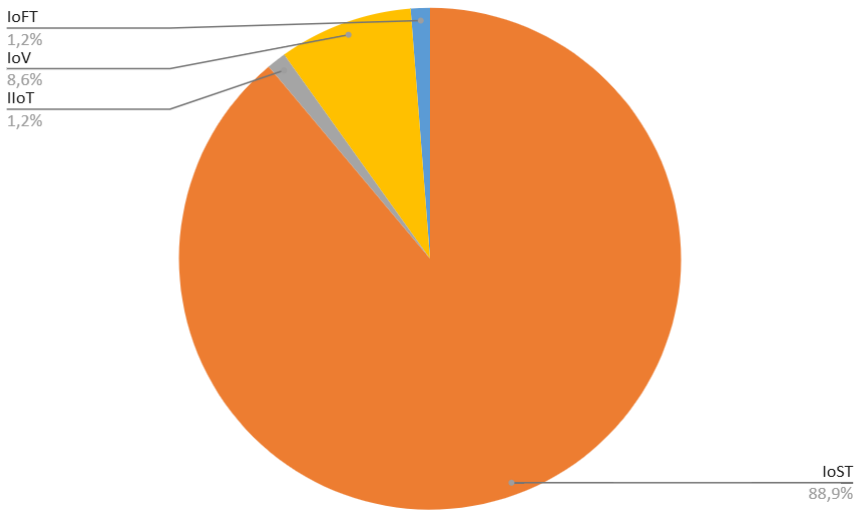


FIGURE 3.5 – Investigation des catégories d'IoT dans les papiers étudiés.

nécessitent une étude plus approfondie.

— L'apprentissage non supervisé a été largement exploré, en particulier à travers les algorithmes de clustering, parmi lesquels le K-means est l'un des plus utilisés pour les applications IoT. Cependant, cet algorithme présente une limitation significative : il nécessite une détermination préalable du nombre de clusters, noté K. Cette exigence peut poser un problème majeur dans les environnements IoT, où le nombre optimal de clusters n'est souvent pas connu a priori. Cette incertitude peut entraîner des performances de clustering sous-optimales, compromettant ainsi l'efficacité des applications IoT.

— En ce qui concerne l'apprentissage par renforcement, les recherches sur le Q-learning restent limitées. Bien que cette approche puisse offrir des solutions prometteuses pour le clustering dynamique et l'adaptation en temps réel des modèles, il existe encore un besoin crucial de développer des algorithmes plus robustes et efficaces, qui exploitent pleinement le potentiel du Q-learning dans les environnements IoT.
— L'apprentissage profond représente une opportunité considérable dans le domaine du clustering, mais son potentiel n'a pas encore été pleinement exploité. Les approches basées sur l'apprentissage profond pourraient permettre de mieux saisir les structures complexes des données IoT, tout en adaptant dynamiquement le nombre de clusters en fonction des données disponibles.

3.5 Conclusion

Dans ce chapitre, nous avons passé en revue les protocoles de clustering basés sur le ML pour les réseaux IoT. Ces protocoles, bien que variés dans leur approche, partagent tous un objectif commun : améliorer l'efficacité des réseaux en optimisant la gestion des ressources et en réduisant la consommation d'énergie. En présentant une taxonomie des algorithmes de clustering basés sur le ML, nous avons classifié ces techniques. Cette taxonomie permet non seulement de mieux comprendre les différentes approches existantes, mais aussi de mettre en lumière les avantages et inconvénients spécifiques de chacune d'entre elles.

Dans le chapitre suivant, nous aborderons nos contributions, qui explorent trois axes du ML : non supervisé, supervisé et par renforcement. Ces approches sont conçues pour améliorer la performance des réseaux IoT en offrant des solutions plus robustes et adaptatives pour le clustering, la gestion de la charge et l'optimisation des ressources.

PARTIE II : Contributions

Chapitre 4

Clustering du réseau IoT : schéma croisé K-means et génétique pour le clustering de capteurs dans l'Internet des objets

4.1 Introduction

Les WSNs sont l'une des émanations très répandues des réseaux IoT, largement utilisés dans diverses applications IoT (voir la section 1.4.1, page 10). Dans ces réseaux, la gestion efficace des ressources, en particulier de l'énergie, est primordiale en raison des contraintes matérielles des capteurs.

Le clustering s'impose comme une solution clé pour optimiser la communication entre les capteurs et prolonger la durée de vie du réseau en regroupant les nœuds en clusters, réduisant ainsi les coûts de transmission et facilitant la gestion des données. Les approches de ML non supervisées sont particulièrement puissantes pour réaliser ce clustering, car elles permettent de détecter des structures ou des schémas cachés dans l'organisation du réseau sans avoir besoin de connaissances préalables ou de données historiques étiquetées (voir la section 2.6, page 28). Cet aspect est essentiel dans l'environnement des WSNs, où les données étiquetées sont souvent inexistantes.

L'un des algorithmes de clustering basés sur le ML les plus connus est celui de K-means. Cette technique de clustering, déclinée sous différentes variantes, a retenu l'attention de plusieurs chercheurs pour diverses applications de réseaux sans fil (voir la section 3.3.1, page 41). Cependant, l'efficacité de cette technique dépend d'un en-

semble de paramètres qui doivent être ajustés avec précision. Le nombre de clusters à former reste toutefois un paramètre problématique de cet algorithme. Pour surmonter cet inconvénient, différents chercheurs ont adopté un certain nombre de méthodes pour déterminer le nombre optimal de clusters. Dans [149], les auteurs ont adopté le clustering par recherche rapide et identification des pics de densité (CFSFDP) [157] ainsi que des techniques d'estimation de densité par noyau (KDE) [158] pour estimer le nombre de clusters le plus approprié. Le CFSFDP identifie efficacement les centres des clusters en localisant les pics de densité. La précision du CFSFDP dépend de la justesse de l'estimation de la densité des nœuds. L'estimation de la densité par noyau est une méthode non paramétrique qui évalue la répartition des données en utilisant des fonctions de densité de probabilité, ce qui la rend utile pour analyser des distributions complexes. Cependant, la KDE s'exécute au moment de la requête, ce qui n'est pas adapté aux grands réseaux. Le travail de [153] étend l'utilisation de la méthode du coude (Elbow method) [164] avec une approche dynamique qui s'adapte aux changements du réseau. Cependant, la méthode du coude nécessite de localiser manuellement le point d'inflexion. La méthode des statistiques d'écart (gap statistics) [164] est utilisée par [165]. Il s'agit d'une méthode largement employée pour déterminer le nombre optimal de clusters. Elle consiste à évaluer une métrique d'erreur prenant en compte la somme des carrés intra-cluster par rapport au nombre total de clusters. Cependant, cette méthode peut conduire à une sous-estimation ou à une surestimation du nombre de clusters en raison des centroïdes initialisés aléatoirement. Certains auteurs ont utilisé des variantes de K-means, telles que l'algorithme X-means, qui suit un processus itératif d'évaluation des résultats de clustering à chaque étape pour déterminer le nombre optimal de clusters [63, 138, 137]. Les auteurs de [139] ont introduit une fonction de fitness dans X-means pour décider de diviser ou non un cluster. Cependant, X-means nécessite toujours la spécification d'une plage pour le nombre de clusters. De plus, l'algorithme divise les clusters sur la base du critère d'information bayésien (BIC), qui doit également être spécifié. Récemment, en utilisant la carte auto-organisatrice (SOM) (voir la section 2.7.3, page 32), les réseaux de neurones ont été utilisés avec succès dans [87] pour déterminer le nombre de clusters. SOM est un algorithme de clustering non supervisé avec des caractéristiques simples et très fiables. Cependant, cette méthode dépend de différents paramètres, notamment la taille de la carte et la matrice de poids, afin d'obtenir de bonnes performances. En outre, cette méthode est coûteuse en termes de calcul et donne des résultats différents à chaque exécution.

Récemment, une nouvelle version de l'algorithme de clustering K-means, appelée K-

means non supervisée (U-k-means), a été proposée pour la reconnaissance des formes [166]. Sans aucune initialisation préalable, cette version permet d'identifier des clusters dans des données représentant des concentrations de tailles et de formes variées.

En conclusion, les travaux basés sur la méthode K-means pour les WSNs tendent souvent à favoriser la sélection des nœuds les plus proches du centre du cluster et disposant d'une bonne énergie pour le rôle de CH. Bien que cette approche puisse être efficace dans certains scénarios, elle ne conduit pas toujours à une solution optimale en termes de performances globales du réseau. Par ailleurs, les algorithmes génétiques se sont avérés efficaces pour résoudre ce problème [167]. Plutôt que de se limiter à des critères locaux tels que la proximité géographique et l'énergie, ces approches basées sur les algorithmes génétiques peuvent prendre en compte des combinaisons complexes de caractéristiques, ce qui permet une sélection plus robuste des chefs de cluster.

Ce chapitre présente une extension de notre première tentative d'utilisation des U-k-means pour résoudre le problème de clustering dans les WSNs [5]. De plus, cette approche introduit une méthode originale de sélection des CHs basée sur les algorithmes génétiques, prenant en compte la performance globale du réseau.

4.2 Notions préliminaires

Afin de mieux comprendre la méthode proposée dans ce travail, cette section présente deux concepts fondamentaux : l'algorithme unsupervised K-means et les algorithmes génétiques.

4.2.1 Unsupervised K-means

L'algorithme K-means (voir la section 2.7.1, page 30) ne peut être considéré comme totalement non supervisé, car il nécessite la spécification de plusieurs paramètres critiques pour son bon fonctionnement.

Un routage basé sur K-means présente des opportunités uniques mais fait face à différents défis. Les paramètres critiques de K-means – nombre de clusters, initialisation des centroïdes et mesure de distance – jouent des rôles déterminants dans les performances de la solution proposée.

Nombre de Clusters (K) : Le nombre de clusters détermine le nombre de regroupements de capteurs. Ce paramètre influence en grande partie l'efficacité du routage induit par la clusterisation du réseau sans fil [153]. Un nombre adéquat de clusters

permet de réduire la surcharge de communication en optimisant la gestion des ressources réseau, telles que l'énergie et la bande passante. Un trop grand nombre de clusters peut entraîner une gestion complexe et une augmentation des interférences entre clusters. À l'inverse, un nombre insuffisant de clusters peut surcharger les chefs de cluster, augmentant ainsi les délais et la consommation d'énergie.

Initialisation des Centroïdes : Une bonne initialisation des centroïdes peut minimiser les coûts de communication intra-cluster et prolonger la durée de vie du réseau.

Mesure de Distance : La mesure de distance influence la formation des clusters en déterminant comment les capteurs sont regroupés.

Récemment, Sinaga *et al.* [166] ont conçu l'algorithme de clustering U-k-means. Fortement inspiré de l'algorithme Expectation-Maximization (EM) [168], cet algorithme prend en compte la probabilité qu'un point de données appartienne à chaque cluster et élimine itérativement les classes dont les probabilités d'appartenance sont les plus faibles, jusqu'à ce qu'un état statique soit atteint. Cet état correspond au nombre optimal de clusters. Ainsi, l'algorithme U-k-means permet de surmonter les inconvénients de la méthode K-means classique, comme le montre le tableau 4.1.

Table 4.1 – K-means VS U-k-means

K-means	Unsupervised K-means
- Requiert le nombre de clusters.	- Prend le nombre de points de données comme nombre initial de clusters.
- Requiert l'initialisation des centroïdes.	- Ne requiert pas l'initialisation des centroïdes.
- Sensible au bruit : chaque point de données est assigné de manière binaire à un cluster spécifique. Cela signifie qu'un outlier peut fausser la position du centroïde du cluster auquel il est attribué, car il est traité comme un membre à part entière de ce cluster.	- Moins sensible au bruit : en attribuant des probabilités d'appartenance, l'algorithme permet à un point de données d'avoir une appartenance partielle à plusieurs clusters.
- Peut produire des clusters vides.	- Pas de probabilité de clusters vides : un cluster contient au moins un point de données.
- Ne propose pas de mécanisme intégré pour éliminer les clusters non significatifs.	- Élimine itérativement les clusters avec de faibles probabilités d'appartenance, simplifiant ainsi la structure des données.

Dans le contexte du clustering dans les réseaux sans fil, ces avantages peuvent être particulièrement pertinents.

— U-k-means détermine automatiquement le nombre de clusters au cours du processus itératif, éliminant ainsi la nécessité de le spécifier à l'avance. Cela est particulièrement utile dans des réseaux de capteurs où le nombre de clusters est indéterminé et où la topologie peut évoluer.

— U-k-means offre une bonne initialisation des centroïdes grâce à son approche probabiliste. Les probabilités d'appartenance guident la position initiale des centroïdes vers les zones où la densité de capteurs est la plus élevée. Cela permet de former des clusters initiaux compacts et bien répartis, minimisant ainsi les distances intra-cluster. Ainsi, les capteurs communiquent sur de courtes distances à l'intérieur de leurs clusters respectifs, réduisant les besoins en énergie pour la transmission de données. Cela prolonge la durée de vie du réseau et améliore sa robustesse face aux fluctuations environnementales et aux changements dans la densité des nœuds.

— Grâce à son approche probabiliste, U-k-means attribue des probabilités d'appartenance à chaque point de données pour chaque cluster. Les points de données aberrants auront des probabilités d'appartenance faibles pour tous les clusters, ce qui limite leur impact sur la formation des clusters. Ainsi, U-k-means peut identifier et isoler ces bruits sans affecter de manière significative la structure globale du clustering.

— En évitant la formation de clusters vides, U-k-means contribue à maintenir une topologie de réseau optimisée où chaque cluster participe activement au routage des données. Les clusters vides peuvent être préjudiciables, car ils n'ont aucun capteur actif pour participer au routage des données. Cela peut conduire à une inefficacité énergétique et à une surcharge de communication entre les clusters actifs. Au lieu de disposer de plusieurs clusters actifs, on pourrait se retrouver avec moins de clusters surchargés, tandis que d'autres restent vides.

— En éliminant les clusters avec des probabilités d'appartenance faibles, U-k-means minimise la communication inutile à l'intérieur de clusters peu définis ou mal formés. En simplifiant la structure des données, U-k-means permet de définir des clusters plus compacts et homogènes. Cela permet une utilisation plus efficace de l'énergie disponible, prolongeant ainsi la durée de vie opérationnelle des capteurs.

L'objectif principal de l'algorithme U-k-means est de minimiser la fonction objective définie comme suit :

$$J_{U-k-means}(Z,a,\alpha) = \sum_{i=1}^{n}\sum_{k=1}^{c} Z_{ik}||x_i - a_k||^2 - \beta n \sum_{k=1}^{c} \alpha_k \ln \alpha_k - \gamma \sum_{i=1}^{n}\sum_{k=1}^{c} Z_{ik} \ln \alpha_k \quad (4.1)$$

où : $\sum_{k=1}^{c} \alpha_k \ln \alpha_k$ et $\sum_{k=1}^{c} Z_{ik} \ln \alpha_k$ représentent, respectivement, les notions de pénalité et d'entropie. Les termes Z_{ik}, a_k, α, n, c, x_i désignent, respectivement, la matrice d'appartenance de chaque point, les centres de clusters, la proportion (probabilité d'appartenance), le nombre de points de données, le nombre de clusters et le point de données i.

L'algorithme U-k-means comprend les différentes équations décrites ci-dessous.

Équation de mise à jour de la matrice d'appartenance Cette formule permet de calculer la matrice d'appartenance Z_{ik} pour chaque point de données x_i. Elle retourne 1 si x_i est plus proche du centroïde a_k que de tout autre centroïde, en prenant en compte le paramètre de probabilité $\gamma \ln \alpha_k$. Sinon, elle retourne 0.

$$Z_{ik}^{(t+1)} = \begin{cases} 1 & \text{si } ||x_i - a_k^{(t)}||^2 - \gamma^{(t)} \ln \alpha_k^{(t)} = \min_{1 \le k \le c^{(t)}} \left(||x_i - a_k^{(t)}||^2 - \gamma^{(t)} \ln \alpha_k^{(t)} \right) \\ 0 & \text{autrement} \end{cases} \tag{4.2}$$

Équation de mise à jour du taux d'apprentissage γ Cette formule ajuste le taux d'apprentissage γ en fonction du nombre de centroïdes actuels $c^{(t)}$. Le taux d'apprentissage diminue exponentiellement avec l'augmentation de c, ce qui permet une diminution progressive de la variation des centroïdes au fur et à mesure des itérations.

$$\gamma^{(t+1)} = \exp\left(-\frac{c^{(t)}}{250} \right) \tag{4.3}$$

Équation de mise à jour de la probabilité d'appartenance aux clusters Cette équation met à jour la probabilité α_k d'appartenance d'un point de données au k-ième cluster. Elle se compose de deux parties : la première mesure la proportion de points affectés au cluster k, tandis que la seconde ajuste cette probabilité avec β et γ pour assurer une bonne répartition des probabilités.

$$\alpha_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} Z_{ik}^{(t+1)} + \left(\frac{\beta^{(t)}}{\gamma^{(t+1)}} \right) \alpha_k^{(t)} \left(\ln \alpha_k^{(t)} - \sum_{s=1}^{c^{(t)}} \alpha_s^{(t)} \ln \alpha_s^{(t)} \right) \tag{4.4}$$

Équation de mise à jour du paramètre d'apprentissage β Cette formule ajuste le paramètre β en fonction de la variation des probabilités d'appartenance α_k. La valeur minimale entre deux expressions est prise afin de contrôler la variance et d'assurer une répartition équilibrée des points.

$$\beta^{(t+1)} = \min \left(\frac{\sum_{k=1}^{c^{(t)}} \exp\left(-\eta n |\alpha_k^{(t+1)} - \alpha_k^{(t)}| \right)}{c^{(t)}}, \frac{1 - \max_{1 \le k \le c^{(t)}} \left(\frac{1}{n} \sum_{i=1}^{n} Z_{ik}^{(t+1)} \right)}{-\max_{1 \le k \le c^{(t)}} \alpha_k^{(t)} \sum_{k'=1}^{c^{(t)}} \ln \alpha_{k'}^{(t)}} \right) \tag{4.5}$$

Où : $\eta = \min(1, 1/t^{\lfloor a/2-1 \rfloor})$. $\lfloor a/2-1 \rfloor$ représente le plus grand entier qui n'est pas supérieur à a.

Formule de normalisation des probabilités et de la matrice d'appartenance Ces formules permettent de normaliser la liste des probabilités d'appartenance α_k ainsi que la matrice d'appartenance Z_{ik}, respectivement.

$$\alpha_k = \frac{\alpha_k^{(t+1)}}{\sum_{s=1}^{c^{(t+1)}} \alpha_s^{(t+1)}} \tag{4.6}$$

$$Z_{ik} = \frac{Z_{ik}^{(t+1)}}{\sum_{s=1}^{c^{(t+1)}} Z_{is}^{(t+1)}} \tag{4.7}$$

Équation de mise à jour des centroïdes Cette équation calcule le nouveau centroïde a_k pour chaque cluster en prenant la moyenne pondérée des points assignés au cluster k.

$$a_k^{(t+1)} = \frac{\sum_{i=1}^{n} Z_{ik}^{(t+1)} x_i}{\sum_{i=1}^{n} Z_{ik}^{(t+1)}} \tag{4.8}$$

Variation des centroïdes Cette formule mesure le changement entre les positions actuelles et précédentes des centroïdes pour chaque cluster.

$$ch_k = ||a_k^{(t+1)} - a_k^{(t)}|| \tag{4.9}$$

L'algorithme 1 résume les étapes de l'algorithme U-k-means.

4.2.2 Algorithmes génétiques

Les GAs sont une famille d'algorithmes utilisés pour résoudre des problèmes d'optimisation combinatoire en s'inspirant du processus de sélection naturelle observé dans l'évolution des espèces vivantes depuis des milliards d'années.

Dans la nature, la sélection naturelle favorise la survie des individus les mieux adaptés à leur environnement. Ce processus repose sur la reproduction et l'hérédité des caractéristiques des parents transmises aux descendants. Les individus qui possèdent des traits bénéfiques ont plus de chances de survivre et de transmettre leurs caractéristiques à la génération suivante, contribuant ainsi à l'adaptation et à l'évolution des espèces.

De manière similaire, les algorithmes génétiques créent une population d'individus virtuels représentant des solutions potentielles à un problème donné. Chaque individu est évalué en fonction d'une mesure de performance (fitness) qui reflète à quel point il est bien adapté pour résoudre le problème. Les individus les mieux adaptés ont plus

Algorithm 1: Algorithme U-k-means

// Initialisation

Initialiser les paramètres : $\epsilon > 0$, $c = n$, $\alpha_k = \frac{1}{k}$, $a_k = x_i$, $\beta = \gamma = 1$, $ch_k = \text{null}$, $t = 0$

Tant que vrai **faire**

  Mettre à jour la matrice d'appartenance Z_{ik} de chaque point x_i selon l'équation (4.2).

  Mettre à jour le taux d'apprentissage γ selon l'équation (4.3).

  Mettre à jour la probabilité α_k des points appartenant au cluster k selon l'équation (4.4).

  Mettre à jour le paramètre d'apprentissage β avec l'équation (4.5).

  Éliminer les centroïdes avec une faible probabilité : si $\alpha_k^{(t+1)} \leq \frac{1}{n}$, alors retirer k de c.

  Si $((t \geq 60)$ et $(c^{(t-60)} - c^{(t)} = 0))$ **alors**

    Mettre $\beta = 0$.

  Fin si.

  Normaliser les probabilités et la matrice d'appartenance α_k et Z_{ik} selon les équations (4.6) et (4.7).

  Mettre à jour les centroïdes a_k selon l'équation (4.8).

  Calculer la variation des centroïdes ch_k avec l'équation (4.9).

  Si $(\max_{1 \leq k \leq c^{(t)}} ch_k < \epsilon)$ **alors**

    Arrêter l'algorithme.

  Sinon

    $t \leftarrow t + 1$.

  Fin si.

Fin tant que.

de chances d'être sélectionnés pour se reproduire et transmettre leurs caractéristiques (représentées par des solutions algorithmiques) à la génération suivante.

Ce processus de reproduction inclut également des opérations d'exploration, comme la mutation, qui introduit de nouvelles caractéristiques aléatoires dans les solutions. Ces nouvelles caractéristiques peuvent potentiellement améliorer les performances des individus dans leur environnement algorithmique. Ainsi, au fil des générations, les solutions tendent à converger vers des solutions optimales ou proches de l'optimalité pour le problème donné, simulant ainsi le processus évolutif observé dans la nature.

Vocabulaire

Le vocabulaire utilisé dans les GA est défini comme suit :

— **Individu** : une solution potentielle au problème considéré.

— **Population** : un ensemble d'individus.
— **Génération** : un ensemble de populations.
— **Chromosome** : une structure de données représentant un individu. Les blocs de données qui composent le chromosome sont appelés gènes.

Principe

L'application des GAs à un problème combinatoire donné implique les opérations communes suivantes :

Encodage : L'encodage du problème est l'étape fondamentale de la résolution d'un problème à l'aide d'un algorithme génétique. Il existe une multitude de types d'encodage [169].

Population initiale : Une collection de chromosomes est créée en fonction du codage et utilisée comme point de départ des algorithmes génétiques. L'initialisation dépend de la connaissance du problème. En l'absence de toute connaissance du problème, une initialisation aléatoire est préférable afin d'encourager l'exploration la plus approfondie possible de l'espace de recherche [170].

Fonction de fitness : La fonction de fitness décrit comment la qualité d'un individu donné est quantifiée numériquement en fonction de ses attributs génétiques (gènes). L'algorithme d'optimisation vise alors à trouver un individu qui maximise ou minimise la valeur de fitness.

Sélection : L'opérateur de sélection définit les règles permettant aux individus les plus aptes de propager leurs caractéristiques génétiques aux générations futures. Les individus de la population sont sélectionnés pour la reproduction en fonction de leur aptitude. Les individus ayant les scores de fitness les plus élevés ont une plus grande probabilité d'être sélectionnés. Il existe de nombreuses méthodes de sélection [169, 171].

Opérateurs génétiques : Pour explorer l'espace des solutions à la recherche d'optima globaux, nous devons nous diversifier. Les opérateurs génétiques permettent cette diversification. Ils se composent de deux opérateurs principaux : le croisement et la mutation. Les individus sélectionnés sont croisés pour produire une descendance. Les informations génétiques des parents sont mélangées pour créer de nouveaux individus au cours du croisement. Les opérations de mutation sont appliquées à certains individus de la descendance afin de maintenir la diversité génétique au

sein de la population et d'empêcher une convergence prématurée vers des solutions sous-optimales. Une mutation est un changement aléatoire dans un gène qui peut introduire de nouvelles caractéristiques ou améliorer une solution existante. Il existe plusieurs opérateurs de croisement et de mutation [169, 172]. Le choix de l'opérateur dépend de son adéquation au problème.

Remplacement : La population parentale est ensuite remplacée par la descendance résultant du croisement et de la mutation. Ce processus garantit que les individus les mieux adaptés restent présents dans la population tout en permettant l'introduction de nouvelles solutions potentiellement bénéfiques. Plusieurs techniques sont utilisées pour effectuer ce remplacement.

Convergence : Le fonctionnement des GAs est limité par un critère d'arrêt qui peut être basé sur un nombre fixe de générations, un seuil de fitness, la convergence de la population, l'absence de progrès ou une limite de temps. Au cours de l'exécution des GAs, la valeur de fitness obtenue est comparée à celle des générations précédentes, et les chromosomes sont mis à jour en conséquence. À la fin de l'exécution, l'individu ayant la meilleure valeur de fitness est retenu comme solution optimale.

L'algorithme 2 décrit le processus d'un algorithme génétique.

Algorithm 2: Processus d'un algorithme génétique

1: Créer une population initiale aléatoire de taille *PopulationSize*
2: **Tant que** le *StopCriterion* n'est PAS atteint
// Le critère d'arrêt *StopCriterion* peut être basé sur un nombre fixe de générations, un seuil de fitness, la convergence de la population, l'absence de progrès ou une limite de temps.
3: Évaluer chaque individu de la population avec la fonction de fitness
4: Sélectionner les individus pour la reproduction
5: Reproduire les individus sélectionnés pour former une descendance
6: Muter certains individus de la descendance
7: Remplacer la population actuelle par la descendance
8: **Fin tant que**
9: Retourner le meilleur individu de la population finale

4.3 Modélisation

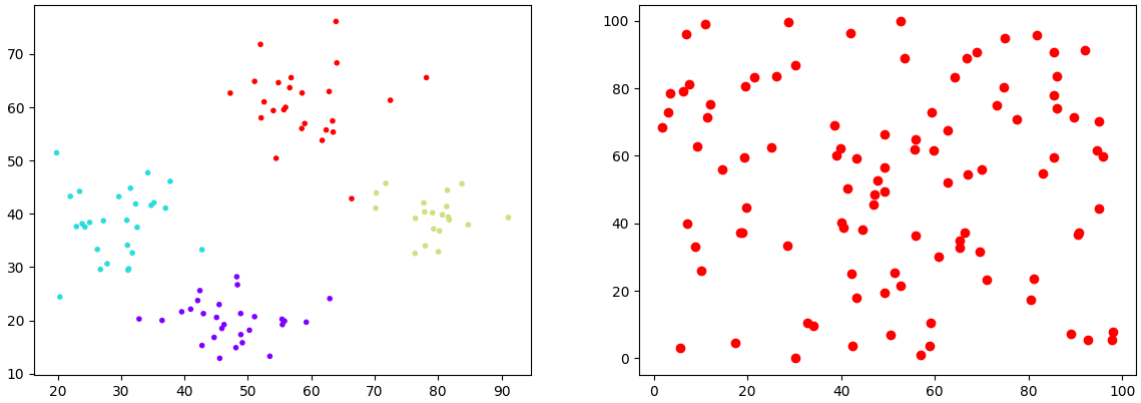
Le modèle du réseau, le modèle de distribution des nœuds ainsi que le modèle énergétique utilisés dans ce travail sont définis comme suit :

4.3.1 Modèle du réseau

Nous supposons que le réseau de capteurs sans fil est déployé sur une zone d'intérêt carrée bidimensionnelle, avec une station de base positionnée à l'extérieur de la zone. Les nœuds du réseau sont statiques et positionnés selon différentes distributions. Les positions des nœuds sont récupérées à partir d'un système de positionnement global (GPS). Les nœuds du réseau ont la même énergie initiale et deviennent dysfonctionnels au-delà d'un certain seuil. La station de base n'a pas de contraintes énergétiques. Les nœuds du réseau peuvent contrôler leur portée de transmission.

4.3.2 Modèle de distribution des nœuds

Nous déployons des capteurs dans la zone considérée en suivant différentes distributions afin de démontrer l'efficacité de U-k-means pour l'identification des clusters. Nous considérons tout d'abord un déploiement basé sur des clusters, afin de démontrer la précision de U-k-means pour la reconnaissance des formes. Ensuite, nous considérons une distribution uniforme afin d'observer comment U-k-means se comporte dans une distribution de probabilité continue. La figure 4.1 illustre les distributions basées sur les clusters et les distributions uniformes.



(a) Distribution basée sur clusters. (b) Distribution uniforme.

FIGURE 4.1 – Modèles de distribution des capteurs.

4.3.3 Modèle énergétique

Les dispositifs de réseaux sans fil offrent plusieurs fonctions, notamment la détection, le traitement des données, la transmission et la réception d'informations. Chacune de

ces fonctions nécessite une certaine quantité d'énergie. Toutefois, la quantité d'énergie la plus importante est consommée lors de la transmission des données [173]. Plusieurs modèles énergétiques ont été proposés dans la littérature [174]. Dans ce travail, nous nous concentrons sur la conservation de l'énergie. Nous considérons donc un modèle de consommation d'énergie tel que l'émetteur a besoin d'une certaine quantité d'énergie pour faire fonctionner l'électronique radio et l'amplificateur de puissance. D'autre part, le récepteur utilise une quantité d'énergie différente pour reconnaître et décoder le signal radio. Plus précisément, pour transmettre un paquet de données de l bits sur une distance d, nous avons besoin d'une énergie d'émission E_T qui, en fonction d'une distance seuil d_0, utilise soit le modèle de l'espace libre, soit le modèle à trajets multiples, comme le montre l'équation (4.10).

$$E_T(l,d) = \begin{cases} E_{elec} \times l + \epsilon_{fs} \times d^2 & \text{si } d < d_0 \\ E_{elec} \times l + \epsilon_{mp} \times d^4 & \text{si } d \geq d_0 \end{cases} \tag{4.10}$$

Où E_{elec} est la puissance dissipée pour faire fonctionner les circuits de l'émetteur-récepteur, ϵ_{fs} représente la consommation d'énergie de la propagation en espace libre et ϵ_{mp} fait référence à la consommation d'énergie de la propagation par trajets multiples.

Pour recevoir le paquet de données de l bits, le récepteur consomme une quantité d'énergie E_R de :

$$E_R = E_{elec} \times l \tag{4.11}$$

4.4 Proposition

Le protocole proposé est un nouvel algorithme hybride combinant U-k-means et l'algorithme génétique (U-k-means GA), développé pour maximiser efficacement la durée de vie du réseau. Cette méthode utilise l'algorithme U-k-means pour le clustering efficace des nœuds et l'algorithme génétique pour la sélection des CHs.

La station de base collecte les positions géographiques des nœuds dans le réseau de capteurs sans fil. Une fois les positions des nœuds collectées, la station de base les normalise à une échelle standard, ce qui facilite le traitement ultérieur des données et garantit la cohérence des mesures. Une fois les positions normalisées, la station de base applique l'algorithme U-k-means pour diviser les nœuds en clusters en fonction de leur proximité spatiale. Une fois les clusters formés, la station de base utilise un algorithme génétique pour sélectionner les chefs de cluster. Une fois le processus de

clustering terminé, les informations sur les clusters et les chefs de cluster sont diffusées dans tout le réseau.

Après la phase de formation des clusters, le protocole fonctionne par cycles, chaque cycle comprenant plusieurs phases : la phase de programmation des communications intra-cluster, la phase de transmission intra-cluster, l'agrégation des données et la phase finale impliquant la transmission des données collectées à la station de base. Les chefs de cluster programment des créneaux de transmission pour les nœuds de leur cluster afin d'optimiser l'utilisation des canaux et d'éviter les collisions. Les nœuds du cluster transmettent les données collectées à leur chef de cluster respectif en fonction des créneaux horaires de transmission programmés. Les chefs de cluster agrègent les données reçues des nœuds de leur cluster avant de les transmettre à la station de base. Les chefs de cluster transmettent les données agrégées à la station de base. La communication intra-cluster à saut unique et la communication inter-cluster à saut multiple sont basées sur le protocole de l'énergie de transmission minimale (Minimum Transmission Energy, MTE) [175].

Pour minimiser la consommation d'énergie liée au processus de clustering, les clusters et les chefs de cluster ne sont changés que lorsque l'énergie du chef de cluster atteint un seuil minimum. Cela permet de réduire les coûts de communication et d'énergie associés au processus de clustering tout en garantissant un fonctionnement efficace du réseau.

La figure 4.2 illustre un organigramme simplifié de l'algorithme proposé.

4.4.1 Clustering des capteurs en utilisant U-k-means

Dans l'algorithme proposé, l'algorithme de clustering non supervisé K-means est appliqué pour diviser le réseau en un ensemble de clusters disjoints. Cependant, avant le clustering, une normalisation des données est effectuée, comme mentionné précédemment. La normalisation des données consiste à réduire ou à augmenter l'échelle des données. Elle améliore les résultats du clustering [176], comme le montre la figure 4.3.

En effet, l'étendue des valeurs dans les données agit comme un poids lors de la sélection de la manière de regrouper les données. Par conséquent, la normalisation de ces valeurs augmente la précision du clustering.

La station de base normalise les coordonnées collectées à l'aide de la méthode Min-Max [6], puis exécute l'algorithme U-k-means, divisant ainsi les nœuds en un ensemble de clusters distincts. Une fois le clustering effectué, la phase GA est déclenchée afin de sélectionner le meilleur ensemble de CHs.

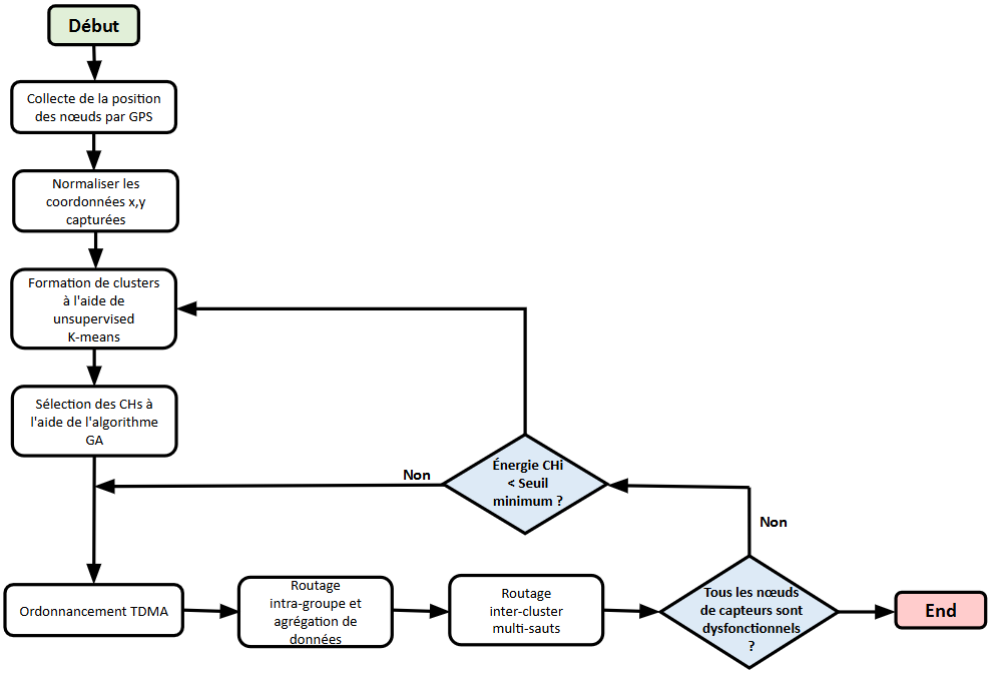


FIGURE 4.2 – Un organigramme simplifié de l'algorithme U-k-means GA.

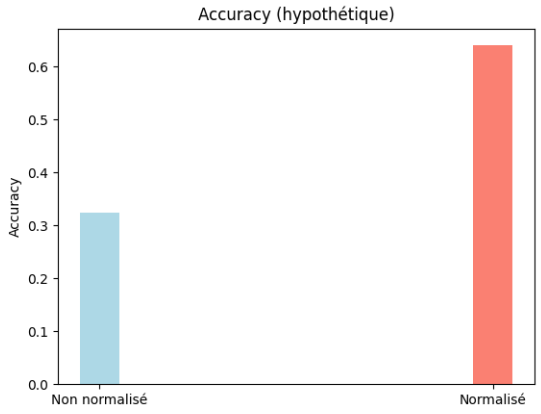


FIGURE 4.3 – Évaluation de la précision du clustering en utilisant la précision hypothétique ($\frac{|\text{Points corrects}|}{|\text{Points totaux}|}$).

L'exécution du processus de clustering avant la sélection des CHs permet d'économiser de l'énergie. En regroupant d'abord les nœuds en clusters, le réseau limite les échanges de messages à l'intérieur de chaque cluster, ce qui minimise la communication entre nœuds distants et évite les transmissions inutiles à l'échelle du réseau entier.

4.4.2 Sélection des CHs basée GAs

Après la phase de construction des clusters, les CHs sont élus à l'aide du GA. La discussion détaillée du processus du GA, y compris le codage des chromosomes, l'initialisation de la population, la fonction de fitness et les opérateurs génétiques, est présentée comme suit :

Encodage

Les algorithmes génétiques ont été adoptés avec succès pour la sélection des CHs dans les WSNs. Le codage binaire est le schéma de codage le plus utilisé, où le chromosome généré représente les nœuds du réseau sous la forme d'une séquence 0/1. Un bit mis à 1 représente un CH et un bit mis à 0 représente un nœud membre ordinaire, de sorte que le processus de sélection peut varier l'identité ainsi que le nombre de CH. Toutefois, dans notre approche, le nombre de CHs est défini par le processus de clustering, ce qui laisse l'identité des meilleurs CHs à définir. Par conséquent, nous utilisons un chromosome décimal de la taille du nombre de clusters, qui englobe l'identifiant des CHs potentiels. La population initiale est créée aléatoirement à l'aide des identifiants des nœuds dans chaque cluster, comme le montre la figure 4.4.

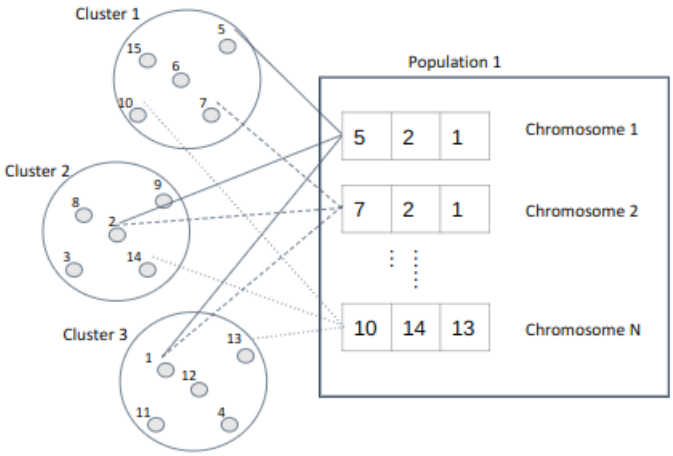


FIGURE 4.4 – Encodage de la population initiale du GA de U-k-means GA.

Fonction de fitness et paramètres

Pour évaluer la qualité d'un individu, la fonction de fitness prend en compte les paramètres calculés en simulant l'exécution du réseau à l'aide de la solution proposée par le chromosome actuel. Ces paramètres incluent :

— **Consommation d'énergie totale du réseau** La consommation d'énergie est considérée comme une préoccupation majeure pour les WSNs. Le choix des CHs a un impact considérable sur la consommation d'énergie. Par conséquent, elle est considérée comme le premier paramètre de notre fonction de fitness.

$$P_1 = ETotal_{intra} + ETotal_{inter} \tag{4.12}$$

Où E_{intra} est la consommation totale d'énergie pour la communication intra-cluster, qui comprend la consommation totale d'énergie des CHs et des membres du cluster. Les CHs consomment de l'énergie pour transmettre le calendrier d'ordonnancement des communications, et pour recevoir les données collectées. Les membres, quant à eux, consomment de l'énergie pour transmettre les données collectées à leur CH correspondant. Le paramètre E_{inter} fait référence à la consommation totale d'énergie pour la communication inter-cluster, il représente l'énergie de transmission totale requise par les CHs pour transférer les données collectées à la station de base.

— **Consommation totale d'énergie des CHs**
Les CHs assurent plusieurs fonctions, à savoir l'ordonnancement, la collecte, l'agrégation et la transmission des données. Par conséquent, leur consommation d'énergie est relativement plus élevée que celle des autres nœuds. Ces nœuds sont donc exposés à une mort prématurée. C'est pourquoi l'énergie totale consommée par les CH est considérée comme le deuxième paramètre de notre fonction de fitness.

$$P_2 = ETotal_r + ETotal_{da} + ETotal_t \tag{4.13}$$

Où $ETotal_r$, $ETotal_{da}$ et $ETotal_t$ représentent respectivement l'énergie totale de réception, l'énergie totale d'agrégation des données et l'énergie totale de transmission des CHs.

— **Écart-type de la consommation d'énergie entre les CHs**
Cette mesure calcule le degré d'uniformité (équité) de la consommation d'énergie entre les CHs.

$$P_3 = \sqrt{\sum_{i=1}^{k}(\mu - ETotal_i)^2} \tag{4.14}$$

Où μ est la moyenne de la consommation totale d'énergie par cluster, $ETotal_i$ est la consommation totale d'énergie du CH i, et k est le nombre de CHs.

— **Distance entre les CHs et la BS en utilisant la communication multi-saut**

Les CHs assurent l'une des principales opérations, à savoir la transmission des données vers la station de base. Généralement, la station de base est située en dehors de la zone de surveillance. Ces nœuds doivent donc communiquer sur de longues distances, ce qui épuise leur énergie plus rapidement. L'utilisation de la communication multi-sauts peut atténuer ce problème. Il est nécessaire de choisir des chefs de cluster qui minimisent les distances d'envoi entre ces derniers. Par conséquent, la distance des CHs par rapport à la station de base, utilisant la communication multi-sauts, est considérée comme le quatrième paramètre de notre fonction de fitness.

$$P_4 = \sum_{i=1}^{k} (min_{ch_i}(d_{ch_i,bs}, d_{ch_i,ch_j} + w_{ch_j,bs})) \tag{4.15}$$

Où ch_i est le CH i, $d_{ch_i,bs}$ est la distance entre le CH i et la station de base, d_{ch_i,ch_j} est la distance entre le CH i et le CH j en utilisant un chemin multi-saut, et $w_{ch_j,bs}$ est la distance directe entre le CH j et la station de base.

L'objectif de l'algorithme GA est d'optimiser (minimiser) la fonction de fitness. Par conséquent, la fonction de fitness est définie par une agrégation linéaire des quatre mesures précédentes :

$$F = \alpha P_1 + \beta P_2 + \gamma P_3 + \delta P_4 \tag{4.16}$$

Où α, β, γ, et δ sont les coefficients de pondération. Comme défini dans l'équation 4.17, ces coefficients sont triés afin de mettre en évidence la contribution significative de chaque facteur.

$$\alpha > \beta > \gamma > \delta \text{ et } \alpha + \beta + \gamma + \delta = 1 \tag{4.17}$$

Les valeurs de pondération des paramètres de la fonction de fitness sont résumées dans le tableau 4.2.

Sélection

La sélection est la base de la reproduction dans les algorithmes génétiques. Dans notre approche une sélection aléatoire est mise en œuvre. Dans notre travail une sélection aléatoire est utilisée, où nous choisissons aléatoirement deux chromosomes de la population. Après chaque itération, une nouvelle génération est créée à l'aide d'opérateurs génétiques à partir des chromosomes sélectionnés. Afin d'éviter la perte des meilleurs chromosomes, la population est triée en fonction du score de fitness et mise à jour par la nouvelle génération, tout en conservant les chromosomes les plus aptes.

Opérateurs génétiques

— **Croisement** Un seul point de croisement est utilisé dans notre approche GA. En fonction du taux de croisement, deux chromosomes échangent des portions séparées par le point de croisement pour obtenir une nouvelle descendance, comme le montre la figure 4.5.

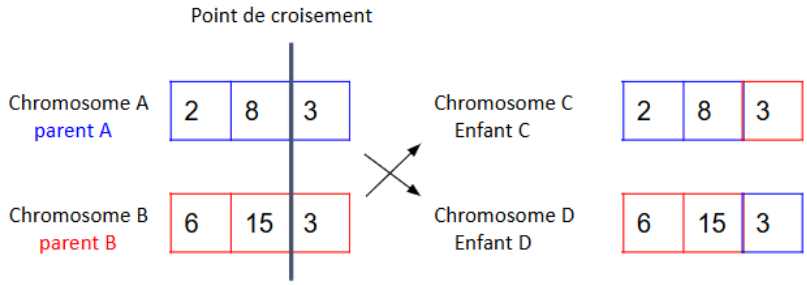


FIGURE 4.5 – Croisement dans le GA de U-k-means GA.

— **Mutation** Nous utilisons une mutation par bit qui affecte un seul gène. En fonction du taux de mutation, des changements peuvent se produire dans les chromosomes et créer ainsi de nouvelles combinaisons. Dans notre approche, un identifiant est modifié par un autre identifiant du même cluster, comme le montre la figure 4.6.

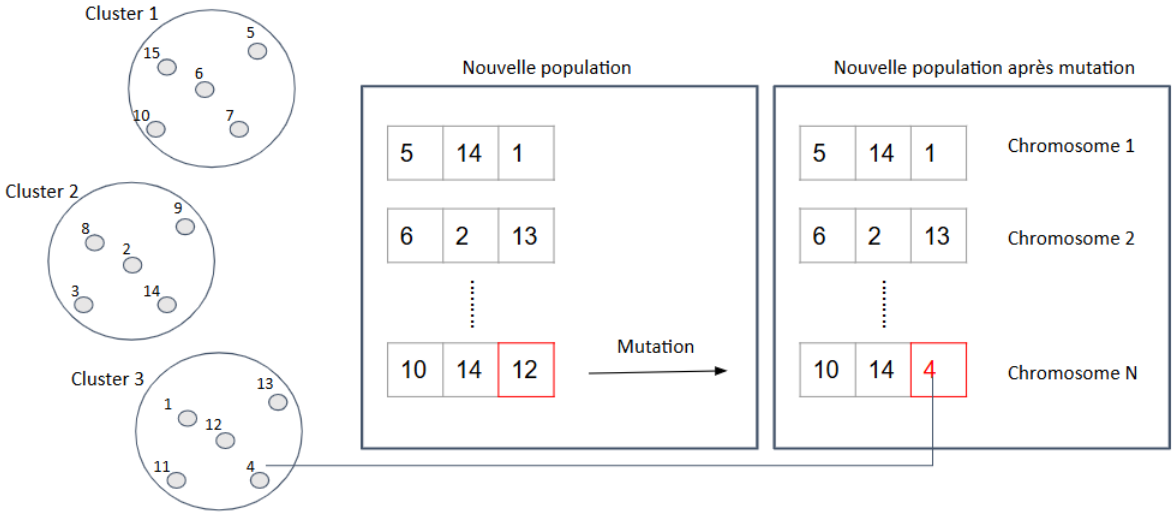


FIGURE 4.6 – Mutation dans le GA de U-k-means GA.

Les valeurs des opérateurs génétiques sont définies dans le tableau 4.2.

Convergence

Notre GA se termine après un nombre prédéfini de générations. À chaque itération, une sélection est effectuée et les chromosomes sont mis à jour en conséquence. À l'issue du processus, le chromosome le plus adapté comprend les identifiants des CHs les plus appropriés.

L'organigramme de l'algorithme GA proposé pour la sélection des CHs est résumé dans la figure 4.7.

4.5 Évaluation des performances

Dans notre expérimentation, nous utilisons l'environnement de simulation PyCharm et Python afin d'évaluer les performances de la proposition.

Les mesures de performance choisies comprennent le nombre de clusters, la consommation d'énergie moyenne et la durée de vie du réseau. L'évaluation du nombre de clusters produits permet une validation relative du nombre de clusters, éventuellement dans la distribution basée sur clusters, où le nombre de clusters est connu a priori. Cette mesure est évaluée au cours de la première phase d'exécution. L'évaluation de la consommation d'énergie permet de déterminer les approches qui assurent une consommation d'énergie efficace, cette dernière étant un problème majeur dans les WSNs. La durée de vie d'un réseau de capteurs sans fil est la période pendant laquelle le réseau peut fonctionner efficacement. Il est essentiel de maximiser cette durée pour maintenir la continuité des opérations et garantir que les capteurs restent opérationnels pendant une période prolongée. Le premier nœud mort (First Node Dead, FND) et le dernier nœud mort (Last Node Dead, LND) sont utilisés pour étayer l'évaluation de la durée de vie du réseau.

Nous comparons notre travail avec des travaux récents trouvés dans la littérature : LEACH-SOM [87], et FFX-means [139], sur la base des mesures de performance. L'évaluation comprend la phase de clustering, la sélection des CHs, l'établissement des tables TDMA, le routage inter et intra-cluster, ainsi que les processus de re-clustering nécessaires.

Nous déployons 100 nœuds dans une zone d'intérêt de 1000 m^2. Nous considérons à la fois des distributions uniformes et des distributions basées sur des clusters. Les mêmes configurations sont utilisées pour les différentes approches. La station de base est située à la position $(50, 150)$.

Les paramètres de simulation sont présentés dans le tableau 4.2.

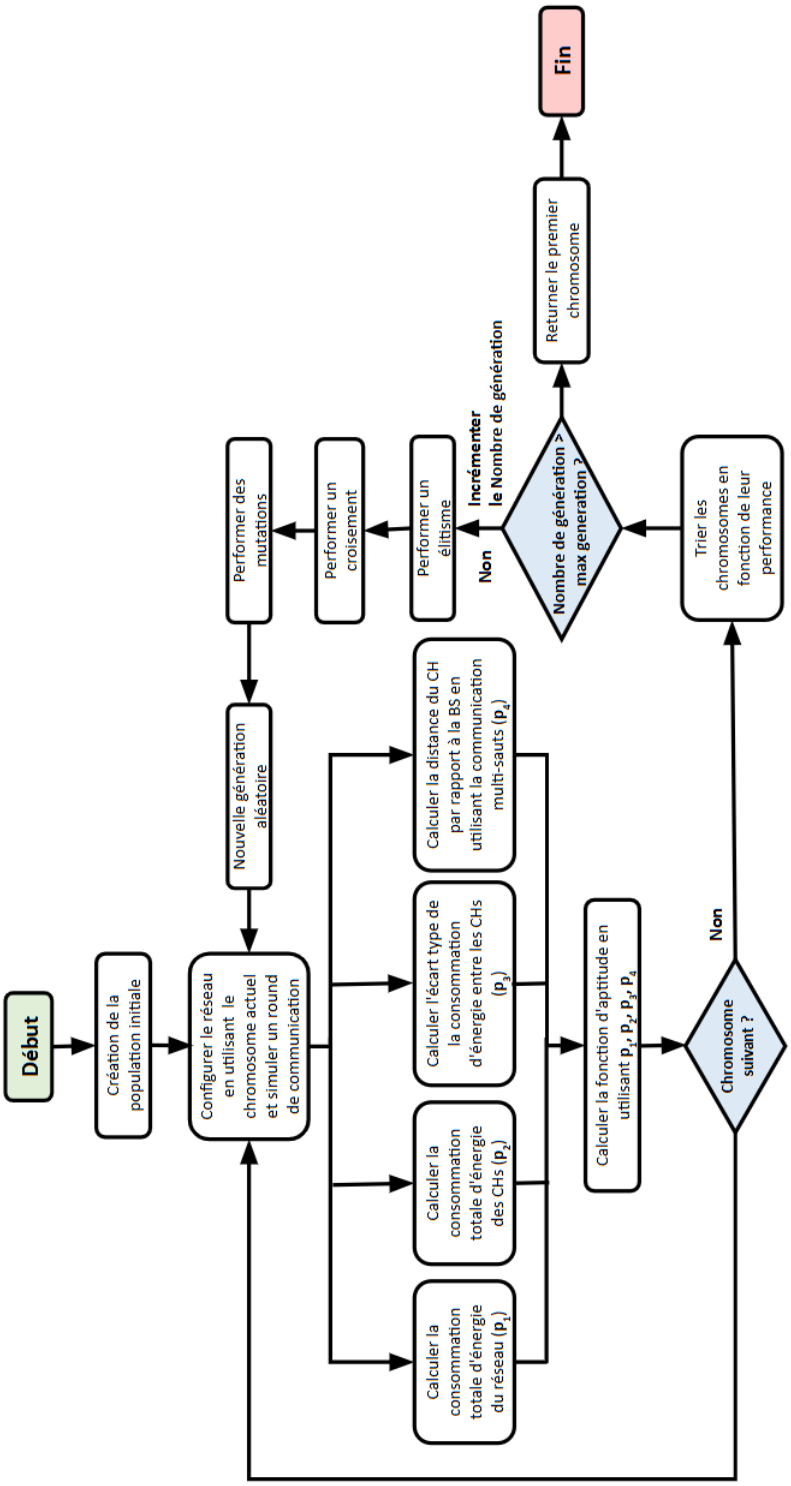


FIGURE 4.7 – Organigramme de l'algorithme GA pour la sélection des CHs dans U-k-means GA.

TABLE 4.2 – Paramètres de simulation

Paramètre	Valeur
Nombre de nœuds de capteurs	100
Taille de la zone de déploiement	100 × 100 m
Position de la station de base	50 × 150 m
Énergie initiale des nœuds capteurs	1 J
Énergie électronique pour l'activité des nœuds	50 nJ/bit
Dissipation d'énergie dans l'espace libre	10 pJ/bit
Dissipation d'énergie lors de la propagation par trajets multiples	0.0013 pJ/bit
Énergie d'agrégation des données	5 nJ/bit
Distance minimale pour déterminer le modèle de transmission	87 m
Nombre de bits à transférer	4000 bits
Seuil d'énergie minimale	0.001 J

Les paramètres de l'algorithme génétique sont résumés dans le tableau 4.3.

TABLE 4.3 – Paramètres de l'algorithme génétique

Paramètre	Valeur
Poids de la fonction de fitness $\alpha, \beta, \gamma, \delta$	$0.4, 0.3, 0.2, 0.1$
Nombre maximum de générations	100
Taux de croisement	50%
Taux de mutation	5%

4.5.1 Distribution uniforme

Nombre moyen de clusters détectés

La figure 4.8 montre le nombre moyen de clusters produits dans la distribution uniforme.

Pour détecter le nombre de clusters, chaque méthode utilise une stratégie différente. FFX-means divise les clusters en fonction d'une fonction de fitness. LEACH-SOM utilise les cartes auto-organisatrices de Kohonen. En revanche, la méthode proposée utilise une probabilité proportionnelle pour déterminer l'appartenance à chaque cluster.

Dans le cas d'une distribution uniforme, la méthode proposée et LEACH-SOM produisent un nombre de clusters élevé. En comparaison, FFX-means produit un nombre moyen de clusters.

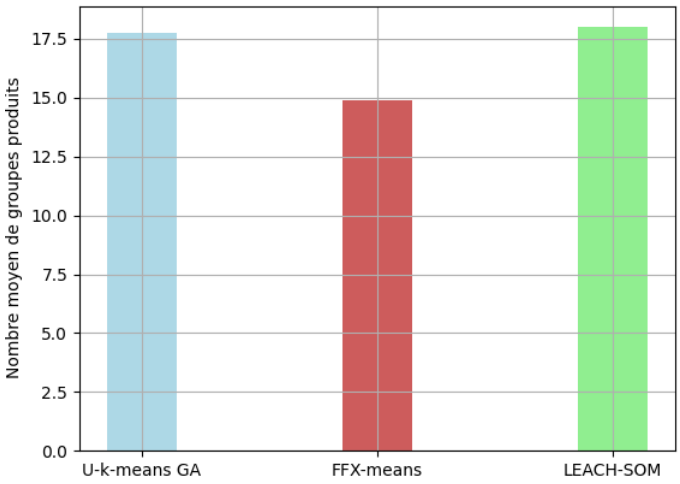


FIGURE 4.8 – Nombre moyen de clusters de U-k-means GA vs FFX-means vs LEACH-SOM selon une distribution uniforme.

Bien que les données d'entrée soient normalisées pour améliorer la précision du clustering dans notre proposition, celle-ci produit un grand nombre de clusters dans la distribution uniforme. En effet, dans la distribution uniforme, les clusters se chevauchent, leurs formes ne sont pas claires et leurs positions peuvent être trop proches les unes des autres, ce qui empêche leur séparation.

Avec FFX-means et LEACH-SOM, un nombre variable de clusters est produit. FFX-means et LEACH-SOM sont affectés par la distribution aléatoire des centroïdes initiaux dans K-means, ce qui entraîne des résultats variables. En outre, le nombre produit par LEACH-SOM est lié à la taille de la carte et aux poids d'interconnexion.

Énergie moyenne consommée

La figure 4.9 montre l'énergie moyenne consommée par chaque approche en tenant compte de la distribution uniforme.

Nous observons que notre proposition améliore efficacement la consommation d'énergie par rapport aux autres approches. Bien que le nombre de CHs soit élevé, la qualité de la sélection des CHs est améliorée grâce à l'algorithme GA. Les nœuds qui améliorent les performances du réseau sont sélectionnés comme CHs. En outre, pour transférer les données collectées à la station de base, les CHs utilisent la communication multi-sauts. Enfin, le processus de re-clustering est effectué lorsque l'énergie des CHs est inférieure à un seuil minimum afin d'économiser l'énergie nécessaire à l'échange de messages entre la station de base et les capteurs pour le processus de re-clustering.

Des pics de consommation d'énergie apparaissent pour les approches FFX-means et

LEACH-SOM. Ce phénomène est lié au processus de re-clustering, qui implique la transmission d'un grand nombre d'informations de contrôle. Cette opération de signalisation et de coordination peut entraîner une augmentation significative de la consommation d'énergie.

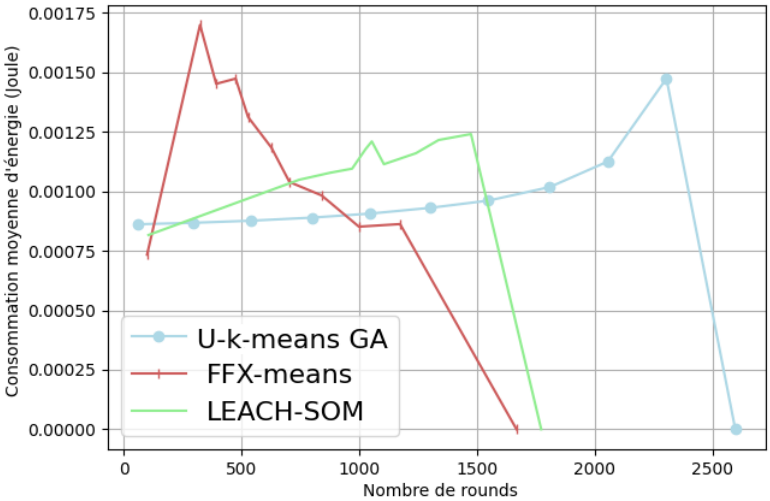


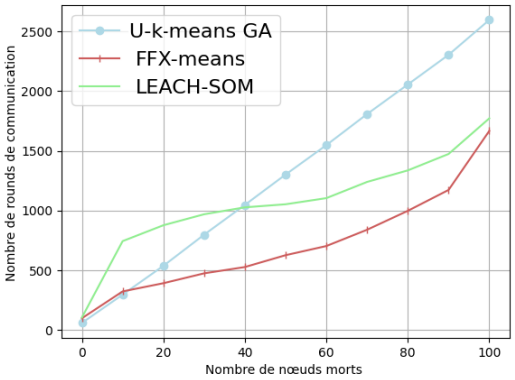
FIGURE 4.9 – Consommation moyenne d'énergie de U-k-means GA vs FFX-means vs LEACH-SOM selon une distribution uniforme.

Durée de vie du réseau

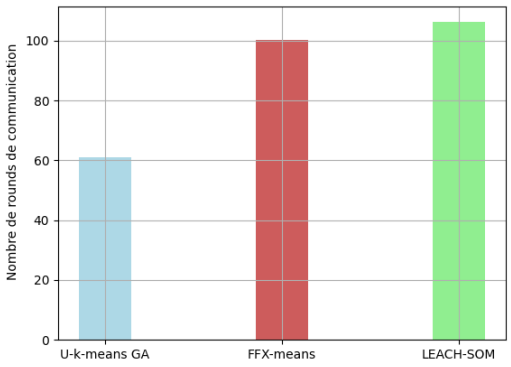
La figure 4.10 illustre la durée de vie du réseau en tenant compte d'une distribution uniforme. Les sous-figures (a), (b) et (c) représentent le nombre de capteurs morts par round, le FND et le LND.

La méthode proposée prolonge efficacement la durée de vie du réseau par rapport aux autres approches. Comme nous l'avons vu précédemment, la méthode proposée consomme le moins d'énergie, ce qui se traduit par une durée de vie du réseau plus longue.

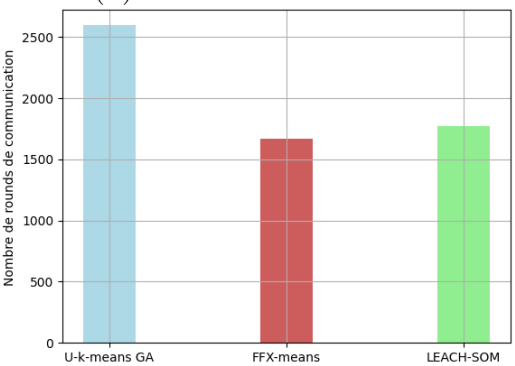
Un FND plus rapide est observé dans notre approche. Cela peut s'expliquer par le fait que notre méthode utilise un routage multi-sauts pour acheminer les données vers la station de base. Les CHs situés à proximité de la station de base consomment davantage d'énergie, ce qui peut entraîner leur dysfonctionnement prématuré.



(a) Nombre de nœuds de capteurs morts au fil du temps.



(b) Premier nœud mort.



(c) Dernier nœud mort.

FIGURE 4.10 – La durée de vie du réseau de U-k-means GA vs FFX-means vs LEACH-SOM selon une distribution uniforme.

4.5.2 Distribution basée sur clusters

Nombre moyen de clusters détectés

La figure 4.11 montre le nombre moyen de clusters produits pour la distribution basée sur clusters.

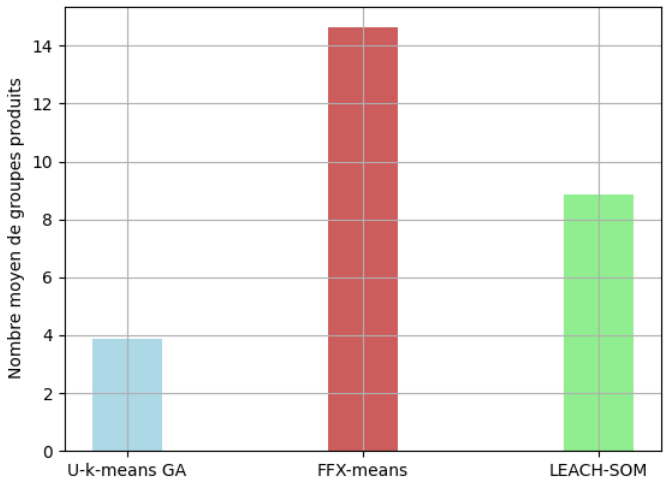


FIGURE 4.11 – Nombre moyen de clusters de U-k-means GA vs FFX-means vs LEACH-SOM selon la distribution basée sur clusters.

Le **tableau 4.4** présente une comparaison des pourcentages d'erreur des trois méthodes : le protocole proposé, FFX-means et LEACH-SOM.

Méthode	**Pourcentage d'erreur (%)**
U-k-means GA	3,1258
FFX-means	265,625
LEACH-SOM	121,875

TABLE 4.4 – Comparaison des pourcentages d'erreur entre U-k-means GA, FFX-means et LEACH-SOM ($\left(\frac{\text{Valeur estimée}-\text{Valeur réelle}}{\text{Valeur réelle}}\right) \times 100$).

Les différentes approches détectent un nombre de clusters moindre par rapport à la distribution uniforme. En effet, dans la distribution uniforme, les clusters se chevauchent, leurs formes ne sont pas bien définies et leurs positions peuvent être trop proches les unes des autres, ce qui complique leur séparation. En revanche, dans la distribution basée sur clusters, les clusters sont définis de manière plus explicite. Le nombre de clusters est même connu a priori, ce qui permet d'évaluer la précision des différents algorithmes.

La méthode proposée présente un pourcentage d'erreur de $3,1258\%$ dans la précision de la détection du nombre de clusters. En revanche, FFX-means et LEACH-SOM ont un pourcentage d'erreur de $265,625\%$ et $121,875\%$, respectivement.

La méthode proposée détecte efficacement le nombre de clusters. Les positions des capteurs sont normalisées, ce qui améliore la précision du clustering. En effet, la normalisation permet de ramener les variables à la même échelle, évitant ainsi que certaines caractéristiques ne dominent le processus de clustering en raison de leurs valeurs plus grandes. En rendant les données comparables, la normalisation aide à produire des clusters plus cohérents et équilibrés, facilitant une meilleure interprétation des résultats.

Comme dans la distribution uniforme, avec FFX-means et LEACH-SOM, un nombre variable de clusters est produit à la fois, à cause de leurs paramètres d'initialisation.

Énergie moyenne consommée

Les figures 4.12 montrent l'énergie moyenne consommée par chaque approche en tenant compte de la distribution basée sur clusters.

Nous observons que U-k-means GA améliore efficacement la consommation d'énergie par rapport aux autres approches. Dans cette distribution une détection plus précise des clusters est réalisée. Comme indiqué précédemment, le nombre de clusters influe sur la consommation d'énergie. Dans chaque cluster, le CH communique avec la station de base, qui est éloignée de la région d'intérêt. Par conséquent, en limitant ce nombre au nombre plus précis de clusters, nous évitons de sélectionner des nœuds CH non nécessaires qui doivent communiquer sur de longues distances. En outre, les clusters détectés sont relativement serrés, ce qui réduit la distance d'envoi à l'intérieur des clusters.

Comme dans la distribution uniforme, des pics de consommation d'énergie apparaissent pour les approches FFX-means et LEACH-SOM. Ces pics sont liés au processus de reclustering, qui implique la transmission d'un grand nombre d'informations de contrôle, ce qui entraîne une augmentation significative de la consommation d'énergie.

La consommation d'énergie est minimale pour toutes les approches de distribution basée sur clusters. Par rapport à cette étude, la distribution basée sur clusters est de loin la meilleure pour la consommation d'énergie dans ce scénario. En effet, cette distribution produit plusieurs clusters dont les membres sont relativement proches les uns des autres, en particulier de leur CH, ce qui réduit la consommation d'énergie intra-cluster et, par conséquent, l'énergie globale.

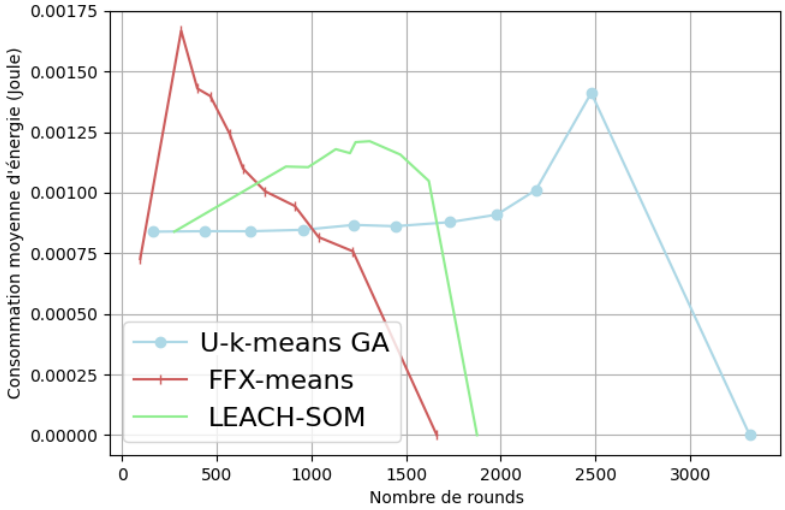


FIGURE 4.12 – Consommation moyenne d'énergie de U-k-means GA vs FFX-means vs LEACH-SOM selon la distribution basée sur clusters.

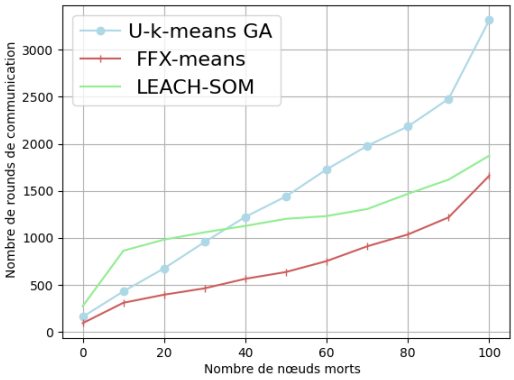
Durée de vie du réseau

La figure 4.13 illustre la durée de vie du réseau en tenant compte d'une distribution basée sur clusters. Les sous-figures (a), (b) et (c) représentent le nombre de capteurs morts, FND et LND.

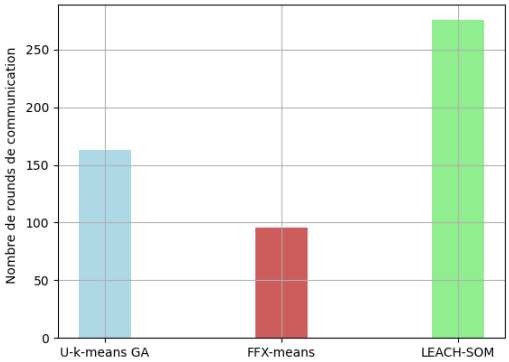
La méthode proposée prolonge efficacement la durée de vie du réseau par rapport aux autres approches du à l'optimisation de la consommation d'énergie.

Par rapport à la distribution uniforme, la durée de vie est particulièrement prolongée dans le cas de la distribution basée sur clusters, où le dernier nœud meurt au 3400$^{\text{ème}}$ round. Dans la distribution basée sur clusters, U-k-means GA consomme le moins d'énergie, les capteurs n'épuisent pas leur énergie rapidement et, ainsi, la durabilité du réseau est améliorée.

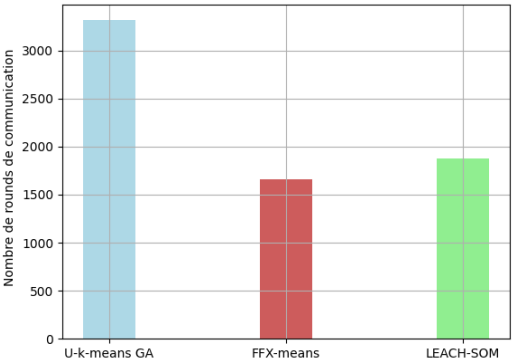
Toutefois, il convient de noter que le choix entre la distribution basée sur clusters et la distribution uniforme dépend des exigences spécifiques de l'application.



(a) Nombre de nœuds de capteurs morts au fil du temps.



(b) Premier nœud mort.



(c) Dernier nœud mort.

FIGURE 4.13 – La durée de vie du réseau de U-k-means GA vs FFX-means vs LEACH-SOM selon une distribution basée sur clusters.

4.6 Conclusion

En conclusion, ce chapitre a mis en lumière l'efficacité des techniques d'apprentissage automatique non supervisées pour le clustering dans les réseaux IoT. Nous avons exploité l'algorithme U-k-means pour regrouper les capteurs en clusters de manière autonome, sans nécessiter de données étiquetées, et intégré les algorithmes génétiques pour optimiser la sélection des nœuds CH. Cette combinaison permet d'améliorer la gestion des ressources, notamment en termes d'efficacité énergétique.

Comparée aux solutions récentes basées sur K-means, notre proposition s'est avérée plus efficace pour détecter le nombre de groupes, réduire la consommation d'énergie et prolonger la durabilité du réseau, pour différentes distributions de nœuds. L'utilisation de différentes distributions de nœuds a mis en évidence la précision de la méthode U-k-means pour la détection du nombre de groupes, en particulier dans le cas de la distribution basée sur clusters, où le nombre de clusters est connu à l'avance.

Dans l'approche proposée, les CHs les plus proches de la station de base consomment plus d'énergie pendant le routage inter-cluster, car ils acheminent les données collectées par les CHs les plus éloignés. Il pourrait être possible d'intégrer des nœuds ordinaires dans le processus de routage inter-cluster. L'inclusion de nœuds ordinaires dans le routage inter-cluster pourrait optimiser l'acheminement multi-sauts des données vers la station de base et réduire la consommation d'énergie élevée des nœuds CH.

Dans le prochain chapitre, nous explorerons les approches par renforcement, qui offrent une perspective innovante pour optimiser les processus décisionnels dans les réseaux IoT.

Chapitre 5

Sélection efficace des CHs dans l'IoT : nouvel algorithme de sélection de chef de cluster basé sur le Q-learning pour les réseaux de l'Internet des objets

5.1 Introduction

Les WSNs constituent l'infrastructure centrale dans le cadre de l'IoT. En effet, ces réseaux jouent un rôle clé en assurant la collecte et la transmission de données issues de divers capteurs déployés pour plusieurs applications (voir page 10). Ces réseaux sont composés de nœuds capteurs à ressources limitées en énergie et en capacité de traitement. La gestion efficace de l'énergie dans les WSNs est donc cruciale pour prolonger la durée de vie du réseau et garantir la fiabilité de la collecte de données. Une technique clé pour atteindre cet objectif est la formation de clusters, où certains nœuds, appelés CH, prennent en charge la collecte et le transfert des données des nœuds voisins vers la base de données principale. La sélection des CHs, cependant, représente un défi majeur, car elle doit minimiser la consommation énergétique tout en maintenant une couverture et une connectivité optimales, tout en s'adaptant aux dynamiques changeantes du réseau, telles que les variations des niveaux d'énergie des nœuds, la densité et la structure du réseau. Les techniques d'apprentissage par renforcement, et en particulier l'algorithme

du Q-learning, se révèlent particulièrement adaptées à ces défis, car elles permettent aux systèmes d'apprendre de l'expérience et de s'ajuster en temps réel aux conditions changeantes du réseau [3]. Les approches de clustering basées sur Q-learning les plus récentes [177, 81, 85] prédéfinissent le nombre de clusters, ce qui limite leur capacité à s'adapter dynamiquement à l'évolution des conditions de ces réseaux. Cette prédétermination ne tient pas compte des multiples facteurs qui influencent les performances du réseau. En outre, ces approches se concentrent principalement sur l'équilibrage de la charge entre les nœuds CH, négligeant d'autres aspects critiques du réseau.

Ce chapitre introduit une nouvelle méthode basée sur le Q-learning pour optimiser le routage dans l'IoT en sélectionnant dynamiquement un ensemble optimal de CHs. Notre fonction de récompense est conçue pour augmenter le débit du réseau, assurer une dispersion efficace des CHs pour une couverture optimale de la région surveillée, promouvoir l'efficacité énergétique et minimiser les délais de communication.

5.2 Notions préliminaires

Afin de mieux comprendre la méthode proposée dans ce travail, cette section présente un concept fondamental qui est : l'algorithme Q-learning.

5.2.1 Algorithme Q-learning

Le Q-learning (voir la section 2.7, page 30) est un algorithme d'apprentissage par renforcement largement utilisé pour former des agents à l'apprentissage d'une politique optimale dans un environnement donné. Une description de son vocabulaire et de son principe est fournie dans les sous-sections suivantes.

Vocabulaire

- **Agent :** Entité qui effectue une action dans l'environnement.
- **Environnement :** Espace dans lequel l'agent opère.
- **État** s **:** Représente la situation ou le contexte dans lequel se trouve l'agent à un instant donné.
- **Action** a **:** Les choix dont dispose l'agent à chaque étape qui déterminent comment modifier l'état actuel.
- **Récompense/pénalité** r **:** Valeur qui représente le succès ou l'échec de l'agent dans son environnement.

— **Q-value $Q(s,a)$:** Valeurs représentant la récompense attendue si l'agent entreprend cette action dans cet état. Ces valeurs sont mises à jour à l'aide d'une Q-fonction pour chaque paire (état, action). Ces valeurs sont stockées dans une table appelée Q-table.
— **Politique π :** La stratégie suivie par l'agent qui consiste en une série d'actions sélectionnées en fonction des estimations de la Q-fonction.
— **Taux d'apprentissage α :** Contrôle la mesure dans laquelle les nouvelles informations modifient les estimations existantes.
— **Facteur d'actualisation γ :** Représente l'importance des récompenses futures par rapport aux récompenses immédiates.

Principe

Le principe fondamental du Q-learning est d'apprendre une fonction Q-fonction, qui fournit une estimation de la valeur d'une action dans un état donné. Cette fonction guide l'agent dans le choix des actions qui maximisent la récompense cumulative à long terme. L'environnement fournit à l'agent des informations sur l'état actuel et attend de lui qu'il agisse en conséquence. L'agent choisit une action en fonction de sa politique actuelle. Il exécute l'action choisie dans l'environnement. En réponse à cette action, l'environnement fournit une récompense, qui est associée à l'état résultant de l'exécution de l'action. L'agent utilise ces informations pour mettre à jour la Q-value associée à la paire état-action $Q(s,a)$ correspondante. Il utilise ensuite la Q-value mise à jour pour ajuster sa politique de sélection des actions. Cette mise à jour de la politique peut impliquer une exploration plus poussée des actions moins explorées ou l'exploitation des actions ayant des Q-values plus élevées, en fonction de la stratégie d'apprentissage choisie, comme le montre la figure 5.1.

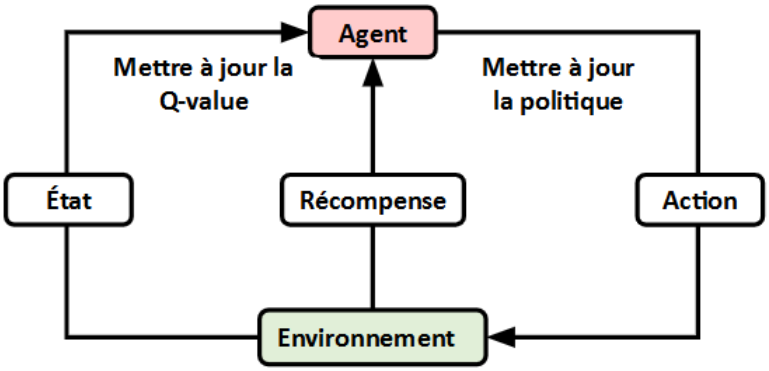


FIGURE 5.1 – Principe de l'algorithme d'apprentissage Q-learning.

L'algorithme 3 résume les étapes classiques de l'algorithme Q-learning.

Algorithm 3: L'algorithme Q-learning

Entrée : États S, actions A, taux d'apprentissage $\alpha \in [0, 1]$, facteur de discount $\gamma \in [0, 1]$, taux d'exploration $\epsilon \in [0, 1]$
Initialiser $Q(s, a) = 0$ pour tous les états $s \in S$ et actions $a \in A$
Pour chaque épisode **faire**
    Initialiser l'état s aléatoirement
        Tant que l'état s n'est pas terminal **faire**
            Choisir l'action a pour l'état s en utilisant une stratégie ϵ-greedy.
            Exécuter l'action a et observer la récompense r et le nouvel état s'
            Mettre à jour la Q-value en utilisant l'équation (5.6).
            Attribuer $s \leftarrow s'$
        Fin tant que
Fin pour
Sortie : Q-table optimisée $Q(s, a)$ pour tous les $s \in S$ et $a \in A$

5.3 Modélisation

Le modèle du réseau, le modèle de distribution des nœuds, le modèle énergétique ainsi que le modèle de trafic utilisés dans ce travail sont définis comme suit :

5.3.1 Modèle du réseau

Le même modèle de réseau utilisé précédemment est considéré (voir la section 4.3.1, page 67), avec une station de base positionnée à l'extérieur de la zone, des nœuds statiques avec la même énergie initiale, qui deviennent dysfonctionnels au-delà d'un certain seuil.

5.3.2 Modèle de distribution des nœuds

Les nœuds sont distribués de manière aléatoire dans la zone de déploiement.

5.3.3 Modèle énergétique

Le même modèle énergétique que précédemment (voir la section 4.3.3, page 67) est utilisé, où un émetteur consomme une quantité d'énergie de :

$$E_T(l,d) = \begin{cases} E_{elec} \times l + \epsilon_{fs} \times d^2 & \text{si } d < d_0 \\ E_{elec} \times l + \epsilon_{mp} \times d^4 & \text{si } d \geq d_0 \end{cases} \tag{5.1}$$

Et un récepteur consomme une quantité d'énergie de :

$$E_R = E_{elec} \times l \tag{5.2}$$

5.3.4 Modèle de trafic

Le trafic de chaque nœud est généré selon une distribution de Poisson [7]. La distribution de Poisson génère des événements et leurs temps d'apparition au cours de chaque round de simulation. Lors de leur apparition, chaque nœud enverra des données. La probabilité de voir k événements (messages) dans un intervalle de temps t est donnée par la distribution de Poisson :

$$P(X = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \tag{5.3}$$

où λ est le taux moyen d'événements par unité de temps.

5.4 Algorithme proposé

Les éléments clés de l'algorithme Q-learning proposé sont les suivants :

— **Agent**

Dans notre approche, la station de base joue le rôle d'agent pour sélectionner les nœuds CHs. Toutes les informations pertinentes sur le réseau sont donc centralisées au niveau de la station de base, ce qui permet d'avoir une vue d'ensemble et de prendre des décisions de sélection plus éclairées et plus efficaces.

— **Environnement**

L'environnement dans lequel l'agent Q-learning opère est le réseau WSNs. Ce réseau se compose de plusieurs nœuds, chacun ayant des attributs tels que la position spatiale, l'énergie disponible et l'état actuel (c-à-d, s'il agit en tant que CH ou en tant que nœud ordinaire).

— **États**

Un état représente la configuration actuelle des nœuds CH dans le réseau sans fil. Plus précisément, il s'agit d'un ensemble de nœuds CH sélectionnés parmi tous les nœuds disponibles dans le réseau à un moment donné. Au départ, l'état est déterminé en sélectionnant au hasard 10% des nœuds vivants en tant que CHs.

— **Actions**

Les actions disponibles pour l'agent sont définies comme suit :

— **« Changer » :** Sélectionner aléatoirement un nœud dans l'ensemble des CHs actuel et le remplacer par un autre nœud.

— **« Supprimer » :** Supprimer de manière aléatoire un nœud de l'ensemble des CHs actuel.

— **« Ajouter » :** Ajouter un nouveau nœud en tant que CH à l'ensemble actuel.

L'action **« Changer »** est favorisée, se manifestant avec une probabilité de 2/3, ce qui encourage une investigation plus approfondie de l'état actuel.

— **Récompenses**

La récompense associée à chaque action est calculée sur la base des performances du réseau après l'application de l'action. Plus précisément, nous simulons le fonctionnement du réseau avec la nouvelle configuration des CHs et calculons une mesure de performance englobant le débit du réseau, la dispersion des CHs, la consommation d'énergie, la consommation d'énergie des CHs, l'écart-type de la consommation d'énergie des CHs et les délais de transmission, comme indiqué dans l'équation (5.4).

$$r_{t+1} = p_1 + p_2 - p_3 - p_4 - p_5 - p_6 \tag{5.4}$$

Où :

— p_1 représente le débit du réseau. L'optimisation du débit du réseau permet de transmettre rapidement et efficacement de grandes quantités de données.

— p_2 représente la dispersion des CHs. Cette valeur est obtenue en calculant la distance entre chaque paire de CHs, puis en renvoyant la distance médiane comme mesure de la dispersion. Une bonne dispersion des nœuds CH dans le

réseau améliore la couverture et la connectivité, qui sont essentielles pour une collecte de données efficace.

— p_3 représente la consommation totale d'énergie du réseau. La consommation d'énergie est une préoccupation majeure pour les réseaux sans fil. La sélection des CHs a un impact significatif sur la consommation d'énergie globale du réseau.

— p_4 représente l'énergie consommée par les CHs pendant la communication inter- et intra-cluster. Les CHs gèrent plusieurs fonctions, notamment la programmation, la collecte, l'agrégation et la transmission des données. Par conséquent, leur consommation d'énergie est relativement plus élevée que celle des autres nœuds. L'optimisation de leur consommation d'énergie permet d'éviter leur défaillance prématurée.

— p_5 représente l'écart-type de la consommation d'énergie des CHs. Cette mesure calcule le degré d'uniformité (équité) entre les CHs. Un écart-type plus faible de la consommation d'énergie entre les nœuds CH garantit une utilisation équitable de l'énergie par tous les CHs.

— p_6 représente le délai de transmission moyen. La réduction du délai de transmission facilite la diffusion rapide des données, ce qui est essentiel pour une prise de décision rapide et des réponses efficaces.

Cette formule de récompense est dérivée des résultats de la simulation de l'exécution d'un round. Avant de calculer la fonction de récompense, les valeurs sont normalisées à l'aide d'une transformation logarithmique. L'équation de la transformation logarithmique peut être définie comme suit :

$$X_{\log} = \log(X + 1) \tag{5.5}$$

Où : X représente la valeur des données originales et $X_{\log}$ représente la valeur des données transformées après application de la transformation logarithmique. L'ajout de 1 à l'intérieur du logarithme permet de traiter les cas où les valeurs peuvent être nulles ou très faibles.

La normalisation des valeurs permet d'éviter que les critères ayant des va-

leurs numériques plus élevées ne prennent le pas sur ceux ayant des valeurs plus faibles. En tenant compte de ces différents facteurs, la fonction de récompense fournit une mesure globale de la performance du réseau. Cependant, même après normalisation, le débit reste le facteur le plus influent. Dans notre contexte des flux de données dynamiques, ceci est être particulièrement avantageux.

— **Mise à jour des valeurs Q :**
Les valeurs de la table Q sont mises à jour à l'aide de l'équation de Bellman comme suit :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \tag{5.6}$$

Cette équation met à jour la valeur de $Q(s_t, a_t)$ en fonction de la récompense immédiate r_{t+1} et de la valeur future attendue $\max_{a'} Q(s_{t+1}, a')$.

— **Critère d'arrêt :**
Dans notre cas, il n'y a pas d'état terminal ; l'objectif est d'optimiser les performances du réseau. Par conséquent, notre méthode comporte deux conditions d'arrêt : l'épisode s'arrête prématurément si le changement dans la table Q devient négligeable, ce qui indique que la politique est presque convergente. Au sein de chaque épisode, les itérations s'arrêtent lorsque le changement des valeurs Q entre les étapes consécutives devient négligeable, ce qui suggère que les mises à jour supplémentaires ne sont plus significatives, ou lorsqu'un nombre maximum d'itérations est atteint. À la fin des épisodes d'apprentissage, l'algorithme sélectionne les CHs finaux en choisissant l'état présentant la meilleure valeur Q. Cet état correspond à une combinaison de CHs qui maximise la fonction de récompense.

La figure 5.2 présente l'organigramme de notre proposition pour la sélection efficace des CHs basée sur Q-learning.

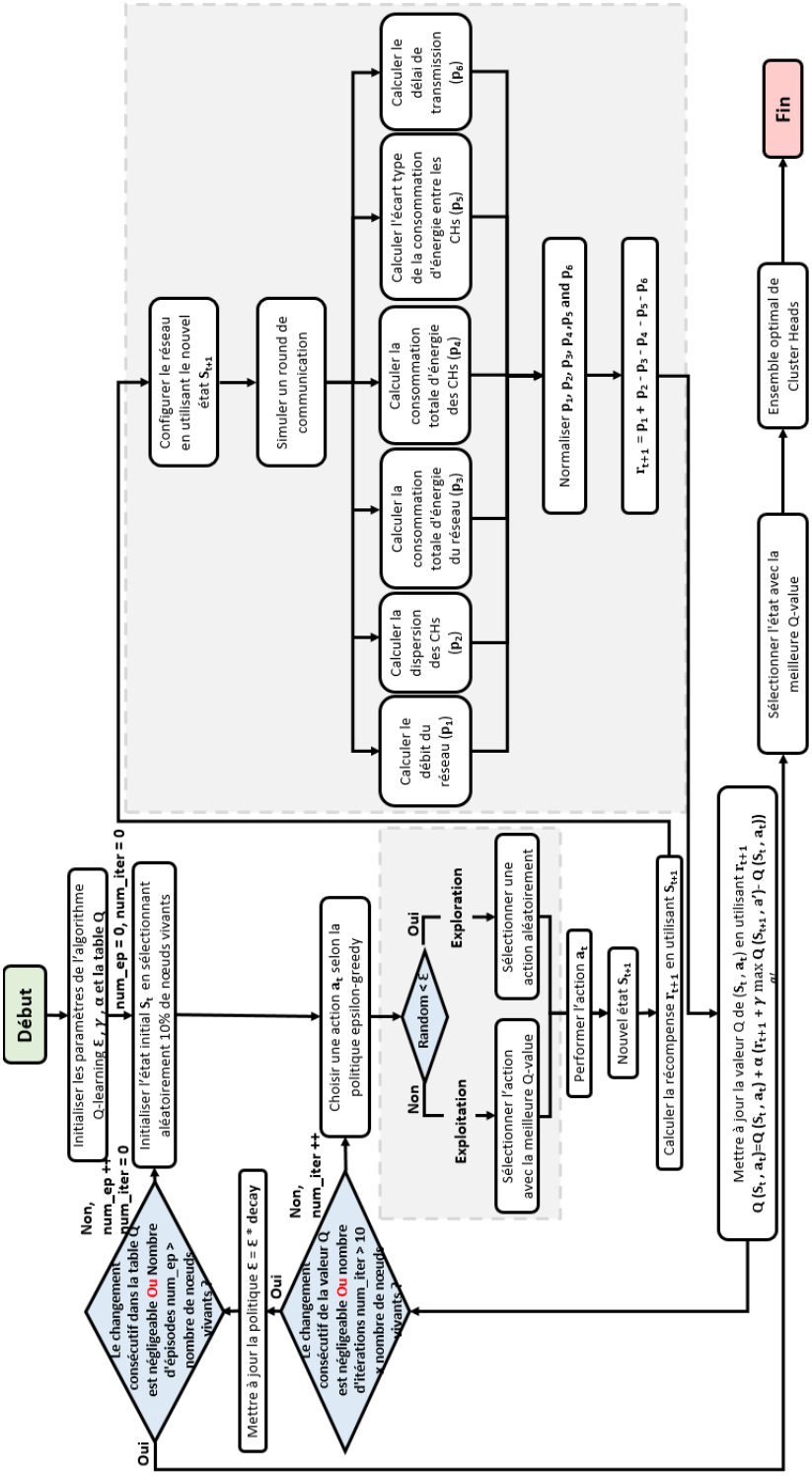


FIGURE 5.2 – Organigramme de la sélection efficace des CHs basée sur Q-learning.

5.5 Simulation et résultats

Dans cette section, nous évaluons les performances de la proposition à travers une série de simulations réalisées avec Python dans l'environnement PyCharm. Nous comparons l'efficacité de notre méthode à celle de l'approche QL_clusters [85], qui vise à former des clusters équilibrés basés sur la densité des nœuds. Pour traiter les cas où il n'est pas possible d'obtenir des clusters parfaitement équilibrés dans cette approche, une limite de nombre maximal d'itérations est imposée. Cette section détaille les paramètres utilisés dans nos simulations et fournit une analyse complète des résultats de performance.

Les paramètres utilisés pour simuler le réseau sont représentés dans le tableau 5.1.

Table 5.1 – Paramètres de simulation.

Paramètres	**Valeurs**
Nombre de nœuds de capteurs	100
Taille de la zone de déploiement	$100 \times 100m$
Position de la station de base	$50 \times 150m$
Énergie initiale des nœuds de capteurs	$1J$
Seuil d'énergie	0,001 J
Période de surveillance (round)	$1000s$
Énergie électronique pour l'activité du nœud	$50nJ/bit$
Dissipation d'énergie en espace libre	$10pJ/bit$
Dissipation d'énergie en multi-trajet	$0.0013pJ/bit$
Distance seuil pour déterminer le modèle de transmission	$87m$
Taux moyen d'événements par unité de temps	10
Nombre de bits à transférer	$4000bits$
Vitesse de transmission	$250kbs$
Seuil d'énergie minimale	$0.001J$

Les paramètres de l'algorithme Q-learning sont résumés dans le tableau 5.2.

Table 5.2 – Paramètres d'apprentissage de l'algorithme Q-learning.

Paramètres	**Valeurs**
Probabilité d'exploration	0.6
Facteur d'actualisation	0.9
Taux d'apprentissage	0.1
Décroissance du taux d'exploration	0.995
Coefficients de pondération (ζ, η, θ, ι, κ, et μ)	0.1, 0.2, 0.1, 0.1, 0.1, 0.4

Nos simulations évaluent le nombre de CHs trouvés, la dispersion des CHs, le pourcentage de messages perdus, le délai de transmission moyen, la consommation d'énergie moyenne du réseau, la consommation d'énergie moyenne des CHs, l'écart type de la consommation d'énergie entre les CHs, et la durée de vie du réseau incluant le premier nœud mort (FND), le demi-nombre de nœuds morts (HND) et le dernier nœud mort (LND).

5.5.1 Nombre de CHs

Le nombre de CHs détectés dépend généralement du protocole de clustering, qui détermine le nombre de nœuds à sélectionner pour gérer les communications. Ce nombre peut varier en fonction de divers facteurs, tels que la taille du réseau, la densité des nœuds, la portée des communications et les objectifs d'optimisation.

La figure 5.3 illustre le nombre de CHs trouvés par la méthode proposée et par QL_clusters. Un nombre variable est indiqué pour notre proposition. Notre approche détecte et ajuste dynamiquement le nombre de CHs au cours des différents rounds de communication.

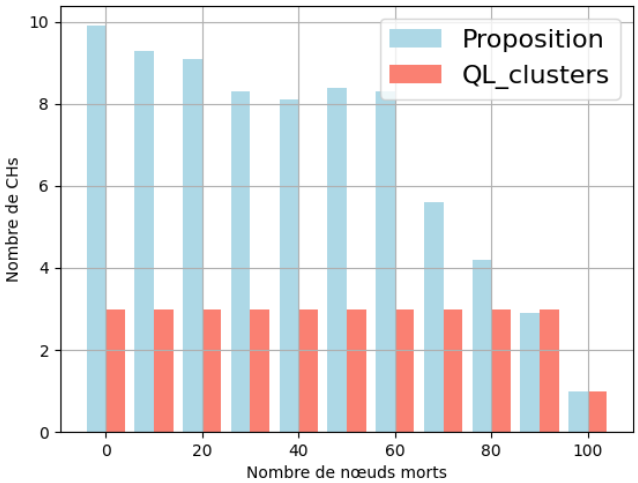


FIGURE 5.3 – Nombre moyen de CHs de notre proposition vs QL_clusters.

5.5.2 Dispersion des CHs

La dispersion des CHs fait référence à la répartition géographique des CHs dans lzone de déploiement du réseau. Un bon paramètre de dispersion indique que les CHs

sont répartis de manière plus uniforme et plus large dans la zone de couverture.

La figure 5.4 décrit la comparaison de la dispersion des CHs entre notre proposition et QL_clusters. La dispersion des CHs varie pour chaque approche. Toutefois, dans l'approche QL_clusters, cette dispersion est principalement due au nombre limité de CHs. Avec seulement trois CHs, chaque nœud CH doit couvrir une plus grande zone et gérer un plus grand nombre de nœuds. Il en résulte une plus grande dispersion géographique, car les chefs de cluster doivent se répartir sur l'ensemble du réseau, ce qui augmente les distances entre eux. En revanche, dans notre approche, la dispersion des CHs bénéficie de l'algorithme Q-learning, qui apprend et ajuste dynamiquement les positions des CHs. Cette capacité d'adaptation permet à l'algorithme d'optimiser activement la distribution des CHs, ce qui permet d'obtenir une répartition efficace sans dépendre uniquement d'un nombre limité de CHs. Ainsi, notre méthode permet d'équilibrer efficacement la couverture et la dispersion des CHs.

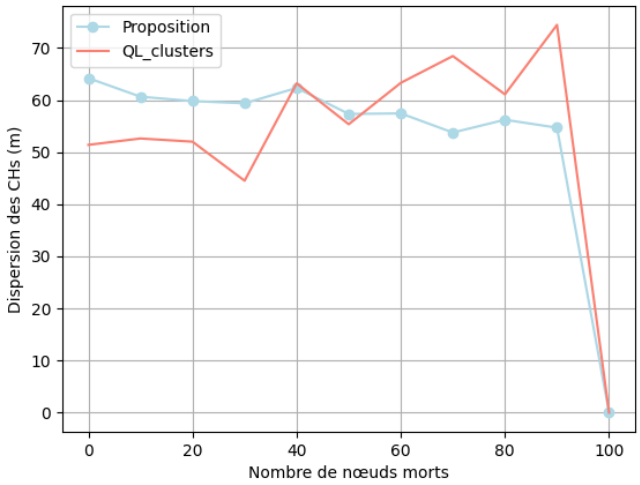


FIGURE 5.4 – Dispersion des CHs dans notre proposition vs QL_clusters.

5.5.3 Pourcentage de messages perdus

Ce paramètre mesure le nombre de messages perdus en raison de la dysfonction des nœuds CH. Il souligne l'importance de ne pas surcharger les clusters afin de couvrir la durée du rounds. Les défaillances des nœuds CH peuvent entraîner une perte de connectivité et une interruption de la transmission des données.

La figure 5.5 décrit le nombre de messages perdus en raison de la dysfonction des nœuds CH entre notre proposition et QL_clusters. Notre approche se démarque claire-

ment en présentant un pourcentage de messages perdus inférieur à celui de QL_clusters. Cette différence substantielle souligne la supériorité de notre approche en termes de gestion des nœuds CH. En effet, en limitant le nombre de clusters à seulement 3, QL_clusters peut entraîner une surcharge des nœuds CH, ce qui augmente le risque de leur dysfonctionnement et, ainsi, la perte de messages. En revanche, notre approche permet une sélection plus souple et plus adaptative des nœuds CH, répartissant la charge de manière plus homogène sur le réseau.

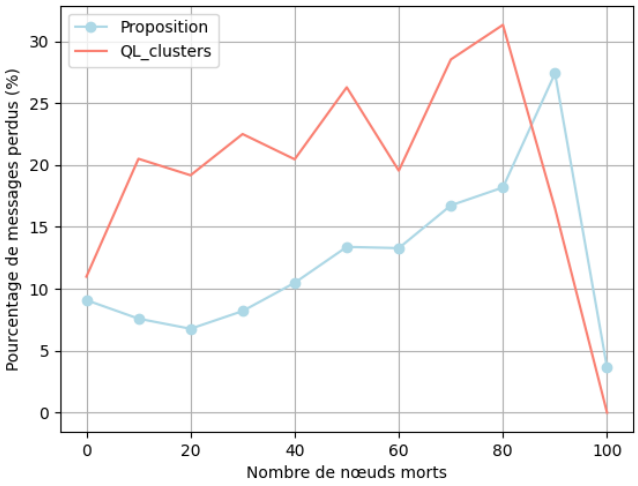


FIGURE 5.5 – Pourcentage de messages perdus dans notre proposition vs QL_clusters.

5.5.4 Taux de réussite

Le taux de réussite indique le pourcentage de messages qui sont transmis avec succès de la source à la destination, sans erreurs ni pertes. Dans notre contexte, les nœuds doivent envoyer les données collectées à la station de base.

La figure 5.6 montre une comparaison du taux de réussite entre notre méthode proposée et celle de QL_clusters. Notre algorithme a un taux de réussite de transmission supérieur à celui de QL_clusters. Cette amélioration substantielle résulte d'une sélection optimisée des CHs, associée à une réduction de la surcharge des clusters. En minimisant le nombre de membres par cluster, notre approche réduit la probabilité de transmissions simultanées, ce qui diminue les collisions et améliore ainsi le taux de réussite des communications.

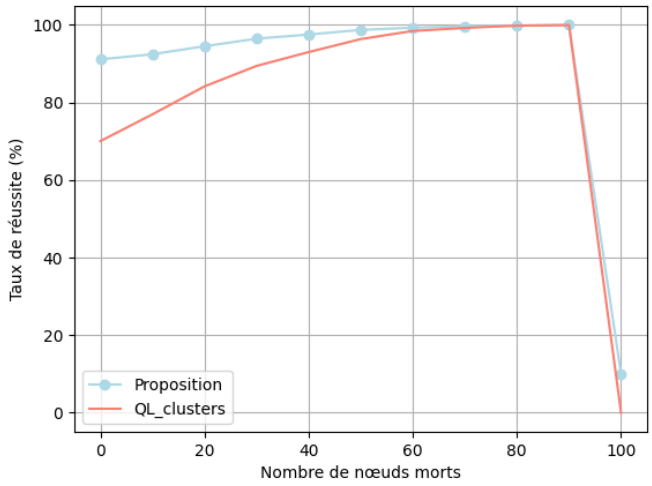


FIGURE 5.6 – Taux de réussite dans notre proposition vs QL_clusters.

5.5.5 Débit

Le débit représente la quantité de données transmises avec succès dans le réseau au cours d'une période donnée. Il inclut les messages qui ont été correctement reçus par la station de base.

La figure 5.7 présente le débit du réseau entre notre méthode proposée et QL_clusters. Notre algorithme affiche un débit supérieur à celui de QL_clusters. Cette amélioration substantielle peut s'expliquer par le pourcentage de perte de messages réduit et le taux de réussite de la transmission plus élevé offerts par notre algorithme.

5.5.6 Délai de transmission moyen

Le délai de transmission moyen est une mesure du temps moyen nécessaire pour qu'un paquet de données soit transmis d'un nœud émetteur à un nœud récepteur dans un réseau. Les délais de transmission doivent être réduits au minimum pour garantir un fonctionnement optimal du réseau.

La figure 5.8 illustre la comparaison du délai de transmission entre notre méthode proposée et QL_clusters. Notre approche présente de meilleures performances en matière de délais de transmission. La distance entre les capteurs et le CH est un facteur clé qui influence les délais de transmission dans les réseaux en clusters. Notre fonction de récompense est spécialement conçue pour minimiser les délais de transmission. En outre, elle réduit la dispersion du CH, ce qui diminue les distances de routage intra-cluster et,

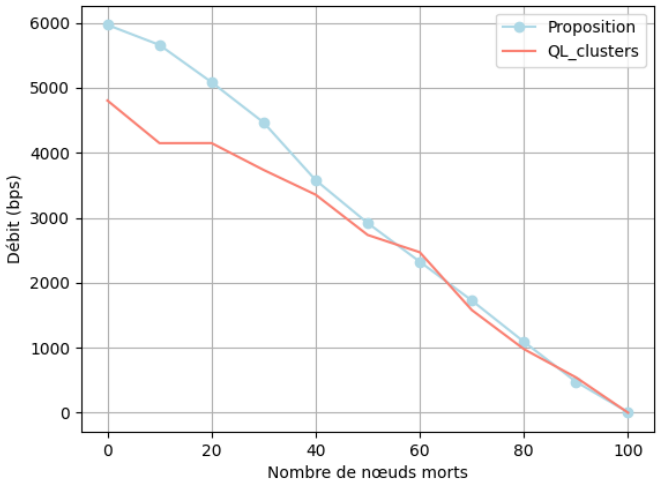


FIGURE 5.7 – Débit noté dans notre proposition vs QL_clusters.

par conséquent, les délais. QL_clusters affiche également de bonnes performances en matière de délais. Cependant, cela peut être dû à une réduction artificielle de la charge de travail sur le réseau. Comme indiqué précédemment, un pourcentage plus élevé de messages perdus est observé pour les QL_clusters, certains messages n'atteignant pas la station de base, ce qui réduit le trafic total à traiter et entraîne des délais plus courts.

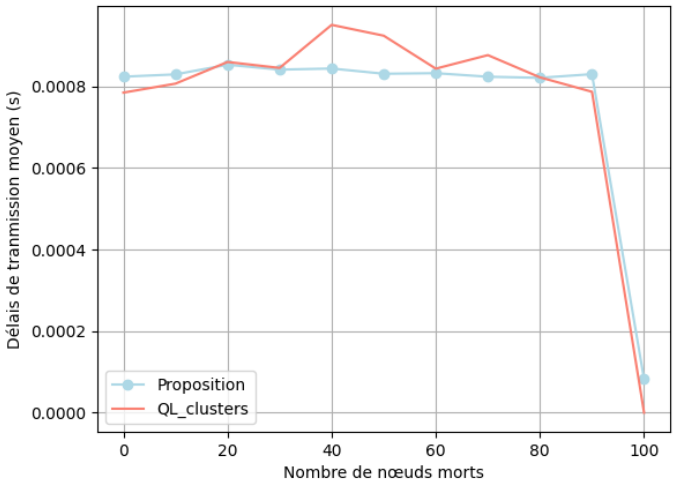


FIGURE 5.8 – Délai moyen noté dans notre proposition vs QL_clusters.

5.5.7 Consommation d'énergie des CHs

En général, la consommation d'énergie d'un nœud CH est plus élevée que celle des autres nœuds du réseau en raison de ses responsabilités accrues en matière de traitement et de communication, ce qui rend l'optimisation de l'énergie essentielle pour éviter une défaillance prématurée de ces nœuds.

La figure 5.9 illustre la consommation d'énergie des CHs de notre méthode proposée par rapport à QL_clusters. Notre approche affiche une consommation d'énergie plus élevée en raison du plus grand nombre de nœuds principaux qui doivent communiquer directement avec la station de base, qui est très éloignée. Chaque CH consomme une quantité importante d'énergie pour transmettre des données sur cette distance.

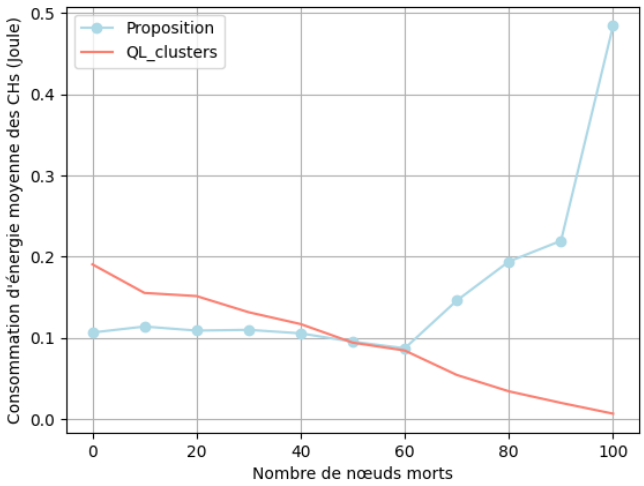


FIGURE 5.9 – Consommation d'énergie des CHs dans notre proposition vs QL_clusters.

5.5.8 Écart-type de la consommation d'énergie entre les CHs

L'écart-type de la consommation d'énergie entre les CHs quantifie la variation ou l'écart entre la consommation d'énergie de chaque nœud CH et la consommation moyenne de tous les nœuds CH d'un réseau. Il nous permet d'évaluer leur équité.

La figure 5.10 illustre l'écart-type de la consommation d'énergie entre les CHs de notre méthode proposée par rapport à QL_clusters. Notre approche présente un écart-type plus faible dans la consommation d'énergie des CHs. Nous équilibrons nos clusters en incorporant l'écart-type de la consommation d'énergie dans notre fonction de récompense. La récompense est calculée en simulant l'activité du réseau dans des conditions de

communication déclenchées par un événement. Cette simulation nous permet d'évaluer avec précision les performances de notre approche dans des environnements dynamiques, en veillant à ce que la récompense reflète la consommation d'énergie et l'efficacité du réseau en temps réel. En revanche, l'approche QL_clusters présente une plus grande variation de la consommation d'énergie, car elle équilibre les clusters en fonction de la densité des nœuds, ce qui est moins efficace dans les scénarios où les communications sont déclenchées par des événements.

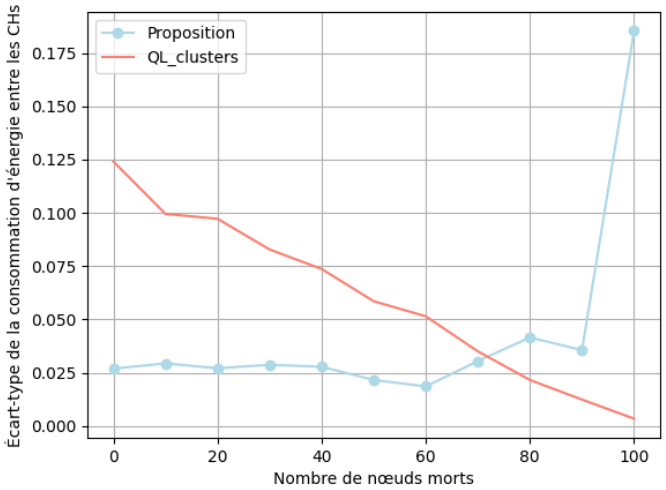


FIGURE 5.10 – Écart-type de la consommation d'énergie des CHs dans notre proposition vs QL_clusters.

5.5.9 Consommation d'énergie du réseau

Les ressources énergétiques sont souvent limitées dans les WSNs, et une gestion efficace de l'énergie est essentielle pour prolonger l'autonomie des nœuds du réseau. En minimisant la consommation d'énergie, notre méthode permet une utilisation plus efficace des ressources disponibles, ce qui garantit une plus longue durée de vie du réseau et une disponibilité continue des capteurs.

La figure 5.11 montre la comparaison de la consommation d'énergie entre notre méthode proposée et QL_clusters. QL_clusters affiche une meilleure consommation d'énergie. Cependant, si QL_clusters semble offrir une consommation d'énergie réduite, cet avantage peut être quelque peu trompeur. Comme indiqué, QL_clusters est associé à un taux de perte de messages plus élevé, ce qui signifie qu'une plus grande proportion de messages n'atteint pas la destination prévue. Cette perte de messages peut faire baisser

artificiellement les mesures de la consommation d'énergie, car ces tentatives de livraison infructueuses entraînent moins de transmissions.

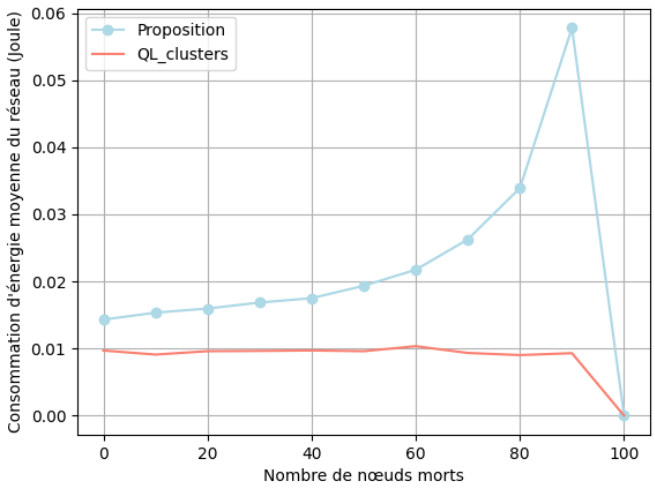


FIGURE 5.11 – Consommation d'énergie des nœuds du réseau dans notre proposition vs QL_clusters.

5.5.10 Durée de vie du réseau

La durée de vie d'un réseau est la période pendant laquelle le réseau reste opérationnel. Le FND, le HND et le LND sont utilisés pour faciliter l'évaluation de la durée de vie du réseau. L'allongement de la durée de vie du réseau garantit une surveillance continue sur une période prolongée.

La figure 5.12 illustre la comparaison de la durée de vie du réseau entre notre méthode proposée et QL_clusters. L'amélioration de l'efficacité énergétique se traduit généralement par une durée de vie plus longue du réseau. C'est pourquoi QL_clusters présente une meilleure durée de vie du réseau. Cependant, notre algorithme présente un retard significatif dans l'apparition de la première défaillance d'un nœud (FND). En effet, notre algorithme accuse un retard significatif dans l'apparition du FND. Généralement, les CHs meurent en premier en raison de la consommation d'énergie plus élevée pour gérer la collecte et la transmission des données pour leur cluster. Notre approche réduit le risque de surcharge énergétique des CHs, retardant ainsi le FND.

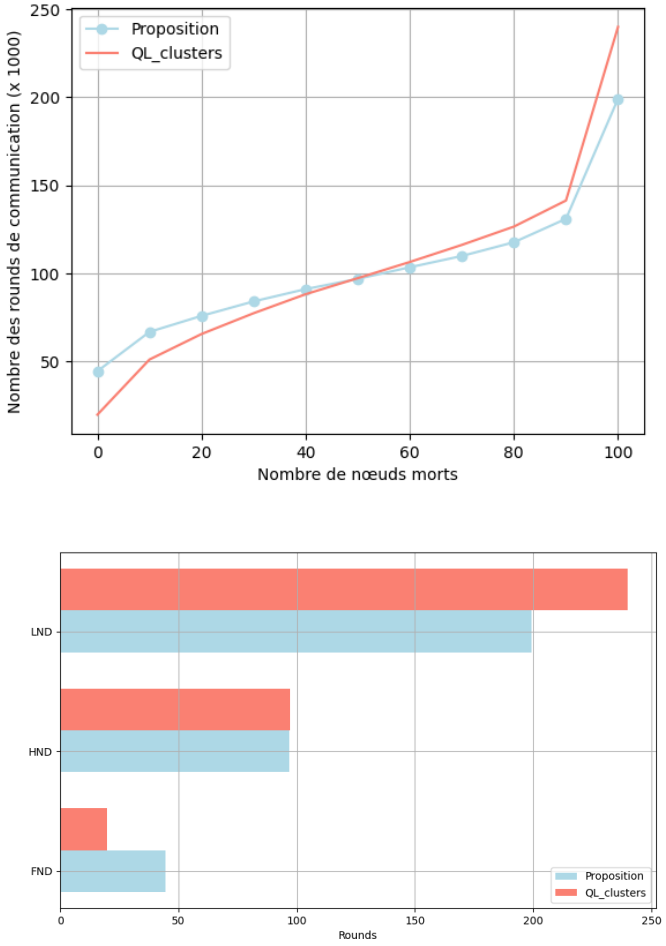


FIGURE 5.12 – Durée de vie du réseau, FND, HND et LND dans notre proposition vs QL_clusters.

5.5.11 Récapitulatif des résultats

La figure 5.13 présente un résumé de la comparaison des performances entre notre méthode proposée et QL_clusters, couvrant la dispersion des CHs, le pourcentage de perte de messages, le taux de réussite, le débit du réseau, le délai de transmission moyen, la consommation d'énergie moyenne des CHs, l'écart type de la consommation d'énergie entre les CHs, la consommation d'énergie moyenne du réseau, FND, le HND et le LND.

Globalement, notre approche se distingue par des performances améliorées sur plusieurs indicateurs clés : le pourcentage de messages perdus, le taux de succès, le débit, l'écart-type de la consommation d'énergie entre les CHs, ainsi que l'occurrence du FND.

Bien que l'approche QL_clusters, qui fixe le nombre de CHs à 3, puisse sembler avan-

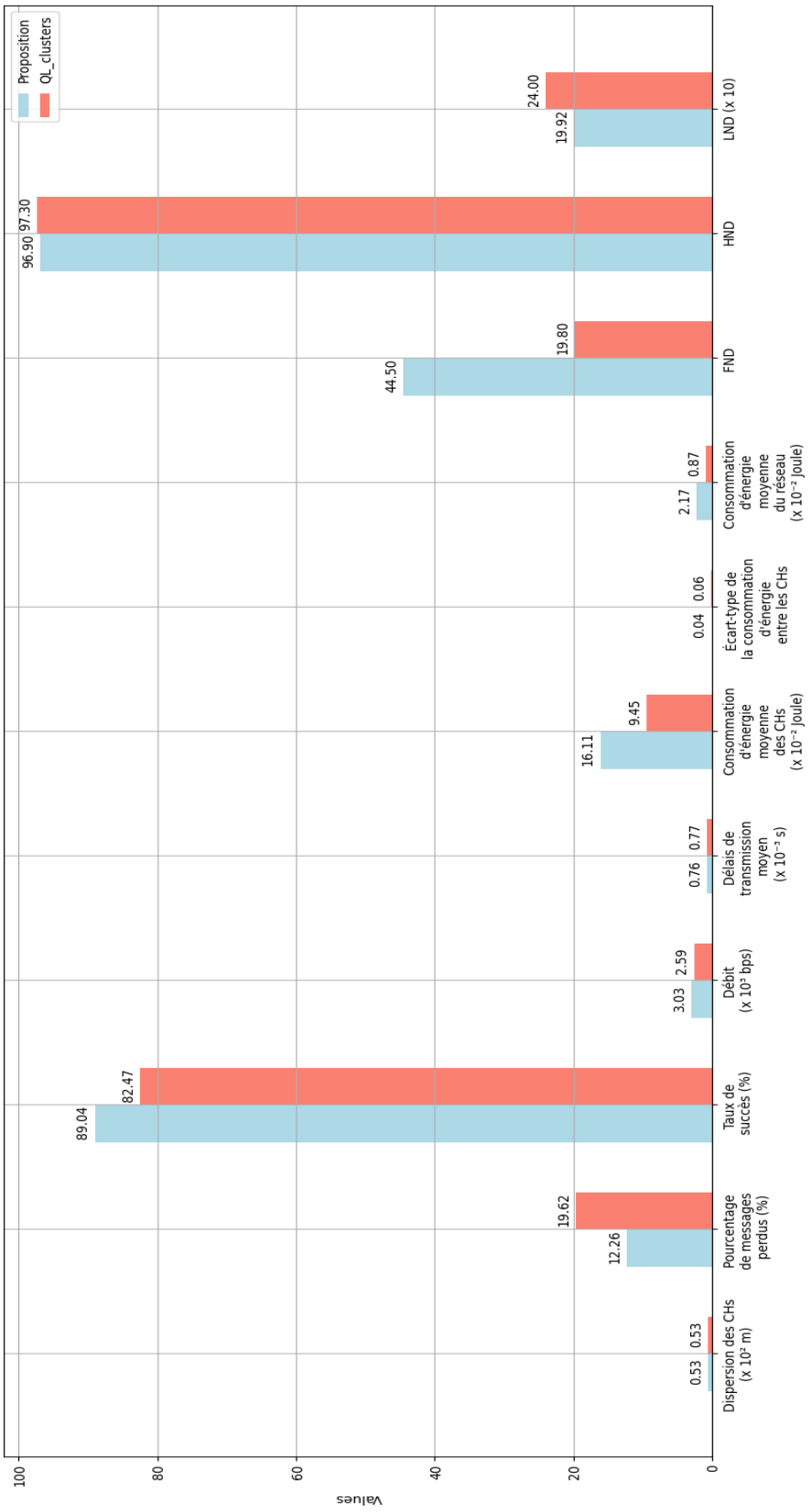


FIGURE 5.13 – Récapitulatifs des résultats de la simulation de notre proposition vs QL_clusters.

tageuse en ce qui concerne certaines métriques d'évaluation, ces résultats peuvent être loin de la réalité. En effet, cette approche conduit à un pourcentage élevé de messages perdus et, du coup, à un débit de communication réduit. La réduction des communications entraîne indubitablement une consommation d'énergie réduite et des délais de communication moindres.

5.6 Conclusion

Ce chapitre a démontré l'efficacité de l'apprentissage par renforcement, en particulier à travers l'algorithme Q-learning, pour optimiser le routage dans les réseaux WSNs. En sélectionnant dynamiquement un ensemble optimal de CHs, nous avons pu répondre à des défis cruciaux relatifs à la performance du réseau.

Bien que notre approche donne de bons résultats en réduisant de manière significative le nombre de messages perdus en raison du dysfonctionnement des CHs, il est important de reconnaître qu'il y a encore de la place pour l'amélioration. En effet, même si notre algorithme minimise les pertes de messages de manière remarquable, il peut y avoir des situations où un certain nombre de messages sont néanmoins perdus. Pour améliorer encore la résilience du réseau, une piste intéressante à explorer serait d'intégrer des techniques prédictives telles que l'apprentissage profond, afin d'anticiper le nombre de messages qui seront produits. En anticipant la charge de travail future du réseau, notre algorithme pourrait sélectionner les nœuds de tête de cluster en fonction de leur capacité à gérer efficacement cette charge supplémentaire, réduisant ainsi davantage les pertes de messages et améliorant la fiabilité globale du système.

Notre proposition a été conçue et évaluée pour un trafic basé sur des événements. Les résultats de la simulation ont démontré l'efficacité de notre approche dans ce contexte. Toutefois, il convient de noter que notre approche présente des améliorations basées sur les expériences réalisées, et que d'autres scénarios doivent être envisagés pour confirmer ses performances. L'évaluation de cette approche dans un contexte de trafic basé sur le temps doit être réalisée. Ainsi, la fonction de récompense doit être modifiée pour mieux répondre aux nouvelles exigences de ce contexte spécifique.

Dans le prochain chapitre, nous explorerons les approches supervisées, qui reposent sur l'utilisation de données historiques pour améliorer les performances du réseau, en exploitant des modèles prédictifs capables de fournir des solutions encore plus précises et adaptées aux besoins spécifiques des réseaux IoT.

Chapitre 6

Équilibrage dynamique de la charge du trafic dans un réseau IoT géré par serveurs edge : approche de clustering dynamique basée sur l'apprentissage profond et les algorithmes génétiques en edge computing

6.1 Introduction

La prolifération rapide des réseaux IoT a entraîné un trafic massif. La surveillance efficace de ce trafic est cruciale pour garantir la sécurité et la fiabilité de ces systèmes [178, 9]. Dans ce contexte, le edge computing apporte un nouveau paradigme pour surveiller et gérer le trafic réseau [179]. Cette technologie place les ressources informatiques, telles que les dispositifs IoT, plus près de la source de données, à la périphérie du réseau. Cette proximité permet de surveiller et de gérer le trafic réseau en temps réel, car les données n'ont pas besoin de parcourir de longues distances pour atteindre un emplacement centralisé à des fins d'analyse. Cette réactivité en temps réel est essentielle pour les applications nécessitant une faible latence, telles que les véhicules autonomes,

l'automatisation industrielle ou l'analyse en temps réel.

Le edge computing répartit le traitement et l'analyse du trafic réseau entre plusieurs dispositifs edge plutôt que de s'appuyer uniquement sur une infrastructure centralisée [180]. Les serveurs edge disposent de l'intelligence et de la puissance de calcul nécessaires pour prendre des décisions locales concernant la gestion du trafic sur le réseau. Ils peuvent appliquer de manière autonome des politiques de trafic, acheminer des données, donner la priorité à certains types de trafic, détecter des anomalies ou des comportements suspects, et identifier les tendances du trafic. Cette prise de décision locale réduit la nécessité d'une communication constante avec un système centralisé, ce qui améliore l'efficacité et réduit l'encombrement du réseau.

L'expansion rapide des réseaux IoT a entraîné une augmentation substantielle du trafic réseau, créant des défis pour les systèmes de edge computing. Une surveillance efficace du trafic est cruciale pour éviter la surcharge de ces ressources, permettant une répartition équilibrée des charges de travail entre les serveurs edge. Cet équilibrage garantit qu'aucun appareil n'est submergé par un trafic excessif, ce qui optimise l'utilisation des ressources de edge computing et empêche les surcharges [181].

Les techniques d'apprentissage automatique profond s'avèrent particulièrement efficaces pour analyser des données historiques, identifier des modèles et prédire les variations futures du trafic. Parmi les différentes architectures de deep learning, les réseaux de neurones à mémoire à long terme (Long Short-Term Memory, LSTM) [182] excellent dans la modélisation des séries temporelles. Leur capacité à capturer les dépendances à long terme dans les données en fait un choix idéal pour anticiper les fluctuations futures du trafic, permettant ainsi une prise de décision éclairée et réactive dans la gestion du réseau.

L'équilibrage de la charge peut s'appuyer sur l'analyse prédictive pour anticiper les modèles de trafic futurs et prendre des décisions proactives. En analysant les données historiques du trafic et en utilisant des modèles prédictifs, il est possible de prévoir à l'avance les pics de trafic. Cette approche proactive réduit la charge globale du système, y compris des facteurs tels que les temps de réponse et la largeur de bande de communication, améliorant ainsi l'efficacité de l'informatique en périphérie.

Le problème de l'équilibrage de la charge entre les serveurs edge est un défi d'optimisation crucial dans les environnements IoT gérés par serveurs edge. Les algorithmes génétiques (voir la section 4.2.2, page 63) sont particulièrement bien adaptés à la résolution des problèmes d'optimisation. En utilisant des mécanismes inspirés de la sélection naturelle, les algorithmes génétiques peuvent générer et évaluer un large éventail de so-

lutions potentielles, converger vers des solutions optimales et s'adapter dynamiquement aux variations du réseau. Ainsi, en employant des algorithmes génétiques, il est possible de parvenir à une répartition optimale de la charge sur les serveurs edge, ce qui garantit un fonctionnement efficace et équilibré du système IoT.

Dans ce chapitre, nous proposons une méthode innovante pour optimiser l'équilibrage de la charge entre les serveurs edge, garantissant une surveillance uniforme du trafic. En regroupant intelligemment les machines, nous veillons à ce que chaque serveur surveille une quantité de trafic équivalente. Grâce aux LSTM, nous sommes en mesure d'anticiper les fluctuations futures du trafic, ce qui nous permet de mieux planifier la répartition de la charge. En complément, nous utilisons un algorithme génétique pour répartir dynamiquement cette charge entre les serveurs en fonction des prévisions fournies par le modèle LSTM.

Cette approche intégrée permet d'améliorer l'efficacité énergétique, la réactivité du système et la robustesse globale du réseau face aux variations imprévisibles.

6.2 Notions préliminaires

Afin de mieux comprendre la méthode proposée dans ce travail, cette section présente un concept fondamental : les réseaux d'apprentissage profond à mémoire à long terme (LSTM).

6.2.1 Réseaux d'apprentissage profond à mémoire à long terme

Les LSTMs sont des architectures de deep learning récurrentes spécifiquement conçues pour traiter efficacement les séquences temporelles. Ils se distinguent par leur capacité à conserver et à exploiter la mémoire à long terme, ce qui les rend particulièrement adaptés aux tâches de prédiction temporelle. Un réseau LSTM est constitué de plusieurs unités appelées cellules de mémoire, disposées en série pour traiter des données séquentielles. Ces cellules sont capables de capturer et de conserver des informations à long terme tout en interagissant avec les cellules voisines.

Chaque cellule LSTM comporte plusieurs composants clés, chacun contrôlé par des portes spécifiques. Les portes et les équations fondamentales d'une cellule LSTM sont définies comme suit [183] :

— Porte d'entrée (i_t) : Cette porte détermine quelle partie des informations d'entrée doit être ajoutée à l'état de la cellule. Elle est définie par l'équation suivante :

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{6.1}$$

— Porte d'oubli (f_t) : Cette porte décide quelle partie de l'état précédent de la cellule doit être oubliée. Elle est représentée par l'équation suivante :

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{6.2}$$

— Porte de mise à jour de la cellule (g_t) : Cette porte génère de nouvelles informations candidates à ajouter à l'état de la cellule. L'équation associée est la suivante :

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{6.3}$$

— Nouvel état de la cellule (c_t) : L'état de la cellule est mis à jour comme suit :

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{6.4}$$

— Porte de sortie (o_t) : Cette porte contrôle quelle partie de l'état mis à jour doit être utilisée comme sortie à l'instant t. Elle est définie par l'équation suivante :

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{6.5}$$

— État caché (h_t) : L'état caché, qui représente la sortie finale de la cellule à l'instant t, est calculé comme suit :

$$h_t = o_t \cdot \tanh(c_t) \tag{6.6}$$

Où :

— x_t : entrée à l'instant t.
— h_{t-1} : sortie à l'instant précédent.
— σ : fonction sigmoïde.
— tanh : fonction tangente hyperbolique.
— W : poids.
— b : biais.

Les poids et les biais sont appris durant la phase d'entraînement du réseau LSTM et ajustés pour minimiser une fonction de perte spécifique, optimisant ainsi les performances du modèle.

La formation d'un réseau LSTM comporte plusieurs phases : la préparation des données, l'entraînement et l'évaluation. Dans la première phase, les données sont prétraitées de manière appropriée. Cela comprend la division de la séquence temporelle et

la normalisation des valeurs. Grâce à un algorithme d'optimisation, le modèle s'améliore progressivement au fil du temps, les poids étant ajustés pour refléter plus précisément la relation entre les entrées et les sorties au cours de la phase d'entraînement. Différents algorithmes d'optimisation peuvent être utilisés [184]. La phase de prédiction d'un modèle LSTM a lieu après l'apprentissage du modèle et permet de prédire de nouvelles données ou des séquences temporelles futures. Lors de cette phase, les prédictions sont comparées aux valeurs réelles afin d'évaluer les performances du modèle. Différentes métriques peuvent être utilisées [185].

L'algorithme 4 résume l'entraînement d'un réseau LSTM.

Algorithm 4: Entraînement d'un réseau LSTM

Charger($data$)
// Préparation des données
Normaliser($data$)
$Train$, $Test \leftarrow$ diviser($data$)
// Initialisation
Initialiser les poids et les biais du réseau LSTM.
Construire(modèle)
// Entraînement
Pour chaque époque d'apprentissage **faire**
  Pour chaque séquence d'entraînement dans $Train$ **faire**
    Calculer la porte d'entrée i_t à l'aide de l'équation (6.1)
    Calculer la porte d'oubli f_t en utilisant l'équation (6.2)
    Calculer la porte de mise à jour de la cellule g_t en utilisant l'équation (6.3)
    Mettre à jour l'état de la cellule c_t en utilisant l'équation (6.4)
    Calculer la porte de sortie o_t en utilisant l'équation (6.5)
    Mettre à jour l'état caché h_t en utilisant l'équation (6.6)
  Fin pour
// Évaluation
Exécuter la prédiction($Test$)
Évaluer(prédictions)
Fin pour

6.3 Jeux de données

Dans ce travail, nous utilisons un nouvel ensemble de données appelé Dataset of legitimate IoT data (VARIoT) [186].

6.3.1 Dataset of legitimate IoT data

VARIoT est un ensemble de données présente le trafic réseau IoT réel généré par des appareils connectés de différents types fonctionnant dans des conditions réalistes. Il contient différents appareils IoT : assistants vocaux, caméras intelligentes, imprimantes connectées, ampoules connectées, capteurs de mouvement, etc.

6.3.2 Attributs

Pour prédire la quantité de flux, nous utilisons les caractéristiques suivantes :

— Src IP & Dest IP : représente les adresses IP de source et de destination des appareils IoT.
— Timestamp : cette caractéristique représente l'horodatage, et peut aider le modèle à prendre en compte les variations temporelles du trafic.
— Flow Duration : cette caractéristique représente la durée de chaque flux, ce qui peut être utile pour prédire le trafic.

Nous construisons l'attribut "nombre de communications", qui fait référence au nombre de sessions de communication ou de connexions observées au cours d'une période donnée, en calculant le nombre de communications associées à chaque adresse IP à l'aide des caractéristiques précédentes. Le nombre de communications dans une fenêtre temporelle donnée indique le volume ou l'intensité du trafic des dispositifs IoT au cours de cette période. Un nombre élevé de flux peut indiquer des périodes d'activité accrue ou de congestion, tandis qu'un nombre plus faible peut indiquer des périodes plus calmes. L'intégration de cette caractéristique permet au modèle de considérer le niveau global de la charge de trafic comme un facteur prédictif.

Ces données sont normalisées indépendamment à l'aide de la méthode Min-Max [6], puis utilisées pour entraîner le modèle.

6.4 Système proposé

Cette section décrit en détail le système proposé, incluant le modèle de système adopté ainsi que l'approche développée.

6.4.1 Modèle de système

Le modèle de système proposé repose sur une architecture intelligente conçue pour optimiser la gestion des réseaux IoT dans le contexte de l'edge computing. Il comprend un serveur central et un nombre de serveurs edge, chacun doté de capacités de traitement dédiées.

Le serveur central agit comme le cerveau du système, intégrant un modèle d'apprentissage profond pour prédire les fluctuations futures du flux de données à travers le réseau IoT et un algorithme génétique pour reconfigurer la structure du réseau. Il est conçu pour anticiper les changements dans le volume de données générées par les objets IoT et ajuster dynamiquement la configuration du réseau en conséquence. En tant qu'équilibreur de charge, ce serveur répartit le trafic réseau entrant entre les différents serveurs. Il agit également comme un gestionnaire de trafic, veillant à ce qu'aucun serveur ne soit submergé par une charge trop importante tandis que d'autres restent sous-utilisés.

Les serveurs edge sont chargés de gérer en temps réel les objets IoT qui leur sont attribués par le serveur central. Chaque serveur dispose de capacités informatiques décentralisées, lui permettant de surveiller le trafic produit par les objets IoT qui lui sont assignés. En équilibrant la charge, les serveurs edge évitent les goulets d'étranglement potentiels et maintiennent des performances optimales.

Pour maximiser les performances du réseau, le serveur central et les serveurs edge coopèrent. Le serveur central informe les serveurs edge des mises à jour prédictives du flux de données et des objets à gérer par chaque serveur. Ces serveurs, à leur tour, gèrent les objets et renvoient des rapports sur les données réelles générées par les objets IoT.

Les composants et l'architecture du système sont détaillés dans la figure 6.1.

6.4.2 Approche proposée

Cette section présente notre système de gestion des flux utilisant l'apprentissage profond dans un réseau IoT avec des serveurs edge. Le système permet d'équilibrer la charge entre différents serveurs en divisant les machines selon les prédictions du trafic.

Prédiction basée LSTM

Dans le cadre de notre approche de clustering dynamique des réseaux IoT dans l'edge computing, nous incorporons la technologie avancée des réseaux LSTM avec l'apprentissage profond pour la prédiction. Cette méthode de réseau neuronal récurrent exploite

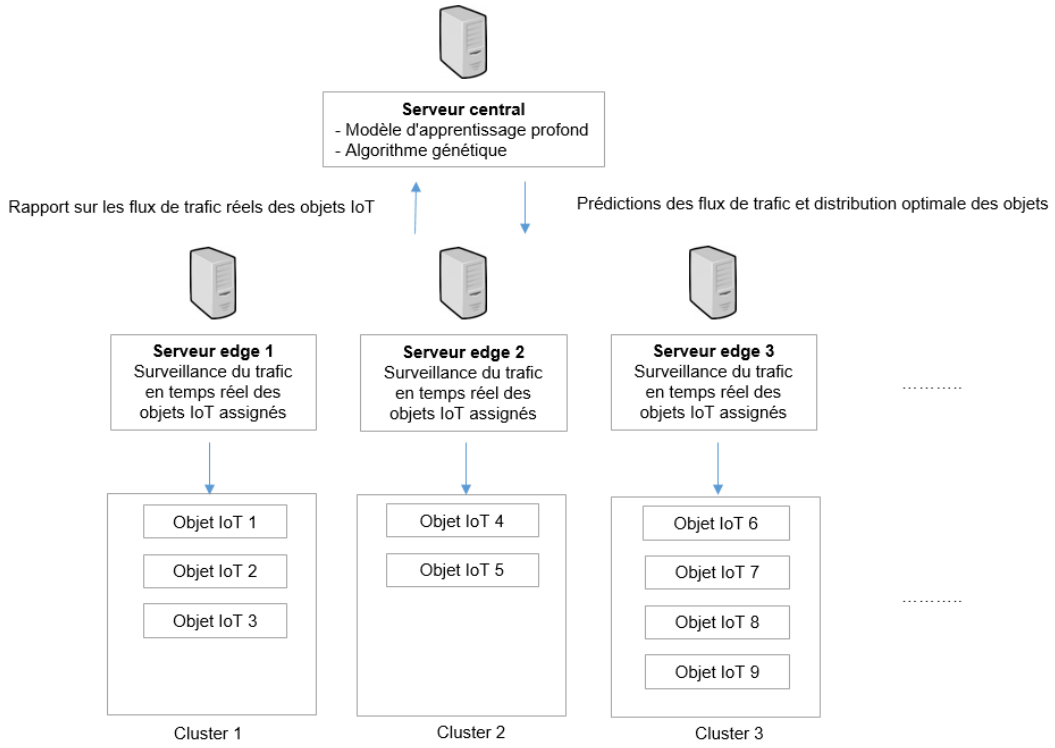


FIGURE 6.1 – Modèle du système proposé pour l'équilibrage de la charge dans un réseau IoT géré par des serveurs edge.

la mémoire à long terme, améliorant ainsi la capacité du modèle à anticiper les futurs schémas de flux de données dans les environnements IoT dynamiques.

— Apprentissage continu
  Une méthode d'apprentissage en ligne est utilisée. Le système recueille des données du réseau IoT, les utilise pour entraîner un modèle de prédiction par apprentissage profond, prédit les flux futurs, répartit la charge sur les serveurs edge en fonction de ces prédictions, évalue les performances par rapport au flux réel, combine les nouvelles données avec les anciennes pour réentraîner le modèle, et répète le processus. Cette méthode permet de conserver les informations des données précédentes tout en intégrant les nouvelles données.
— Régularisation du modèle
  Dans le cadre de notre approche d'apprentissage en ligne, un surajustement peut se produire lorsque de nouvelles données sont ajoutées aux anciennes. Si le modèle s'adapte trop précisément aux nouvelles données sans maintenir une généralisation adéquate, il peut en résulter des performances médiocres sur les données

futures. Les couches dropout (ou couches d'exclusion) sont utilisées pour atténuer ce problème. Elles permettent de régulariser le modèle et de le rendre plus robuste face aux changements de données au fil du temps.

— Introduction de la non-linéarité
Nous utilisons un réseau neuronal profond pour permettre la modélisation de relations complexes et l'extraction de caractéristiques hiérarchiques plus riches. Cependant, même des couches très profondes de réseaux neuronaux agiraient comme des transformations linéaires, ce qui limiterait leur capacité à modéliser des relations complexes dans les données. Les fonctions d'activation introduisent la non-linéarité dans le modèle, ce qui lui permet de capturer et de représenter des motifs et des relations plus complexes. Ce travail utilise la fonction d'activation Rectified Linear Unit (ReLU) [187].

L'architecture du modèle de prédiction proposé est présentée à la figure 6.2.

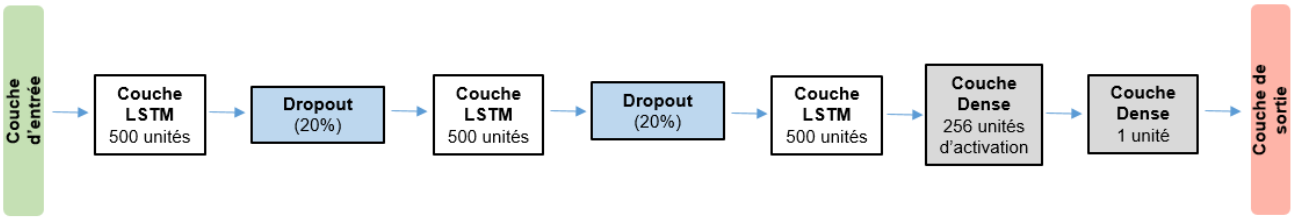


FIGURE 6.2 – Architecture du modèle de prédiction proposé.

Équilibrage de la charge de trafic basé GA

Dans la deuxième partie de notre approche du clustering dynamique des réseaux IoT dans l'edge computing, nous utilisons un algorithme génétique pour assurer l'équilibrage de la charge entre les serveurs. La dynamique de clustering est un résultat direct de la prédiction des flux. Lorsque le modèle LSTM anticipe des changements significatifs dans le flux de données, la structure de clustering des réseaux IoT est adaptée en conséquence. Cela garantit une allocation efficace des ressources, optimisant la gestion des objets connectés et répondant aux exigences spécifiques des environnements dynamiques de l'edge computing. Notre algorithme génétique comprend les étapes suivantes :

— Encodage : Un tableau unidimensionnel est utilisé pour coder les chromosomes. La taille du chromosome est égale au nombre d'appareils IoT. Nous utilisons une représentation décimale pour les chromosomes. Chaque gène a pour valeur un entier s dans l'intervalle $[1 : S]$, où S est le nombre de serveurs edge, ce qui

indique que l'appareil IoT associé est attribué au serveur S. Chaque solution potentielle est représentée sous la forme d'un ensemble de serveurs, chacun avec une liste de machines qui lui sont attribuées, comme le montre la figure 6.3.

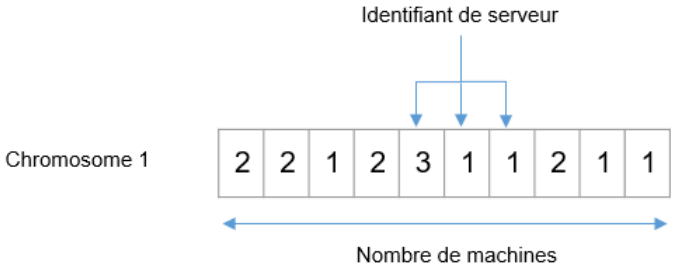


FIGURE 6.3 – Encodage du chromosome.

— Initialisation de la population : Une population initiale d'individus est générée de manière aléatoire. Cela permet de créer diverses solutions potentielles, comme le montre la figure 6.4.

— Fonction de fitness : Chaque individu est évalué en calculant la différence de charge totale entre les serveurs edge. Cette différence est mise en évidence à l'aide de l'écart-type. Une valeur plus faible indique une répartition plus équilibrée des machines entre les serveurs. Ainsi, notre fonction de fitness calcule l'écart-type de la charge entre les serveurs. L'algorithme génétique cherche à minimiser cette valeur, ce qui conduit à une distribution plus équilibrée des machines entre les serveurs. La fonction de fitness est décrite dans l'équation 6.7.

$$F = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{6.7}$$

Où :

— N représente le nombre de serveurs.

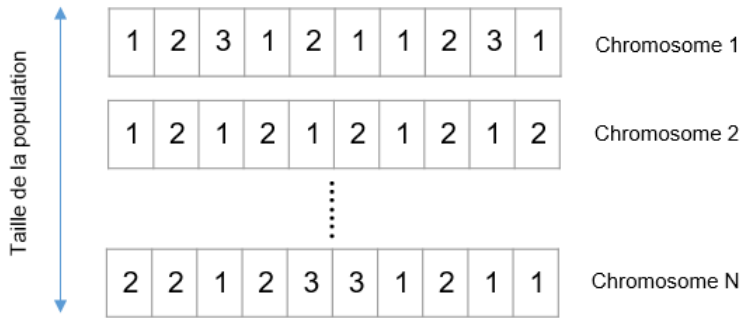


FIGURE 6.4 – Population initiale.

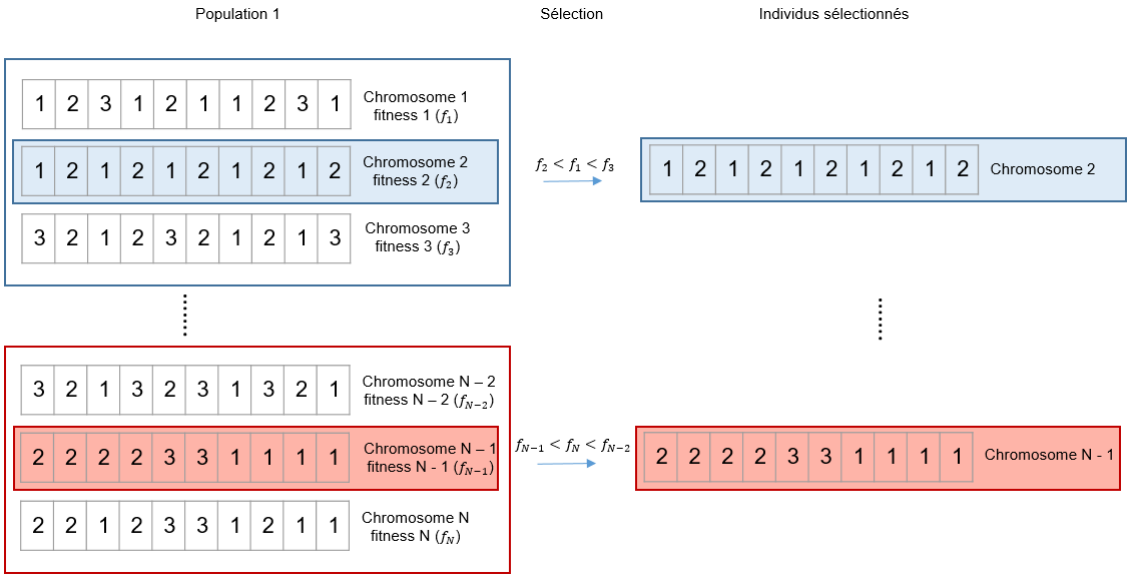


FIGURE 6.5 – Sélection par tournoi.

  - x_i est la charge de chaque serveur.
  - $\bar{x}$ représente la charge totale moyenne.
- Sélection des individus : Les individus les plus performants sont sélectionnés pour la reproduction. Nous effectuons la sélection à l'aide de la méthode de sélection par tournoi. Cette méthode consiste à organiser plusieurs tournois au cours desquels plusieurs individus sont choisis dans la population. Parmi ces individus, celui qui a la meilleure fitness est choisi comme vainqueur du tournoi, comme le montre la figure 6.5.
- Croisement : Les individus sélectionnés sont croisés pour créer une nouvelle génération. Nous utilisons un croisement à point unique, où deux parents sont séparés à un point de croisement aléatoire pour créer deux enfants, comme le montre la figure 6.6.

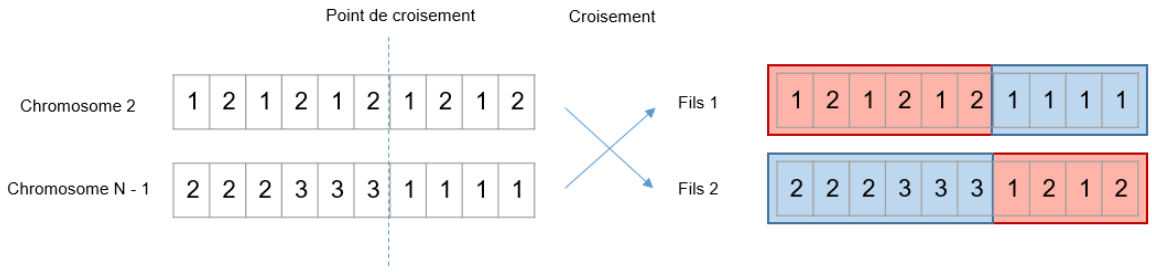


FIGURE 6.6 – Croisement.

— Mutation : Nous utilisons la mutation pour introduire de la diversité dans les solutions candidates. La mutation utilisée est une mutation de base par bit qui affecte un seul chromosome. Ainsi, nous modifions aléatoirement certains individus de la nouvelle génération en changeant aléatoirement le serveur assigné à certaines machines, comme le montre la figure 6.7. La mutation se produit avec une probabilité définie par le taux de mutation, spécifié dans le tableau 6.2.

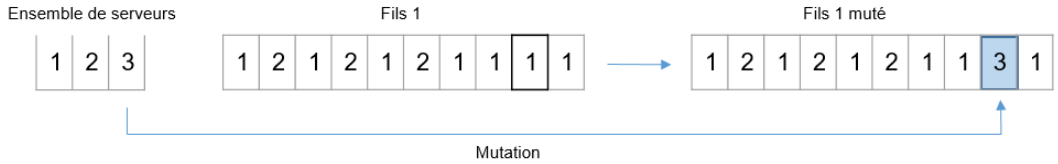


FIGURE 6.7 – Mutation.

— Remplacement : Une nouvelle génération remplace la population précédente, en gardant les individus les plus performants.

La sélection, le croisement, la mutation et le remplacement sont répétés pendant un nombre déterminé de générations. La meilleure solution est déterminée en sélectionnant l'individu dont la valeur de la fonction fitness est la plus faible. Cette solution correspond à une répartition des machines entre les serveurs edge qui minimise l'écart-type de la charge, conduisant ainsi à une répartition équilibrée de la charge.

L'organigramme du système proposé est présenté à la figure 6.8.

6.5 Expérimentation

Nous avons utilisé l'outil de simulation PyCharm pour mettre en œuvre et simuler notre système prédictif dynamique. Comme indiqué précédemment, nous avons opté pour un nouvel ensemble de données, VARIoT [186]. Nous utilisons des données de dates différentes pour représenter des hétérogénéités différentes, assurant ainsi une représentation complète des fluctuations temporelles. Le modèle est évalué en quatre phases de réentraînement, ce qui permet une analyse approfondie de sa robustesse et de son efficacité dans différents contextes temporels. Le premier ensemble de données est utilisé pour pré-entraîner le modèle, tandis que les autres sont utilisées pour évaluer les prédictions et réentraîner le modèle. Trois serveurs edge sont considérés dans notre expérimentation.

Les paramètres du modèle LSTM sont résumés dans le tableau 6.1.

Où :

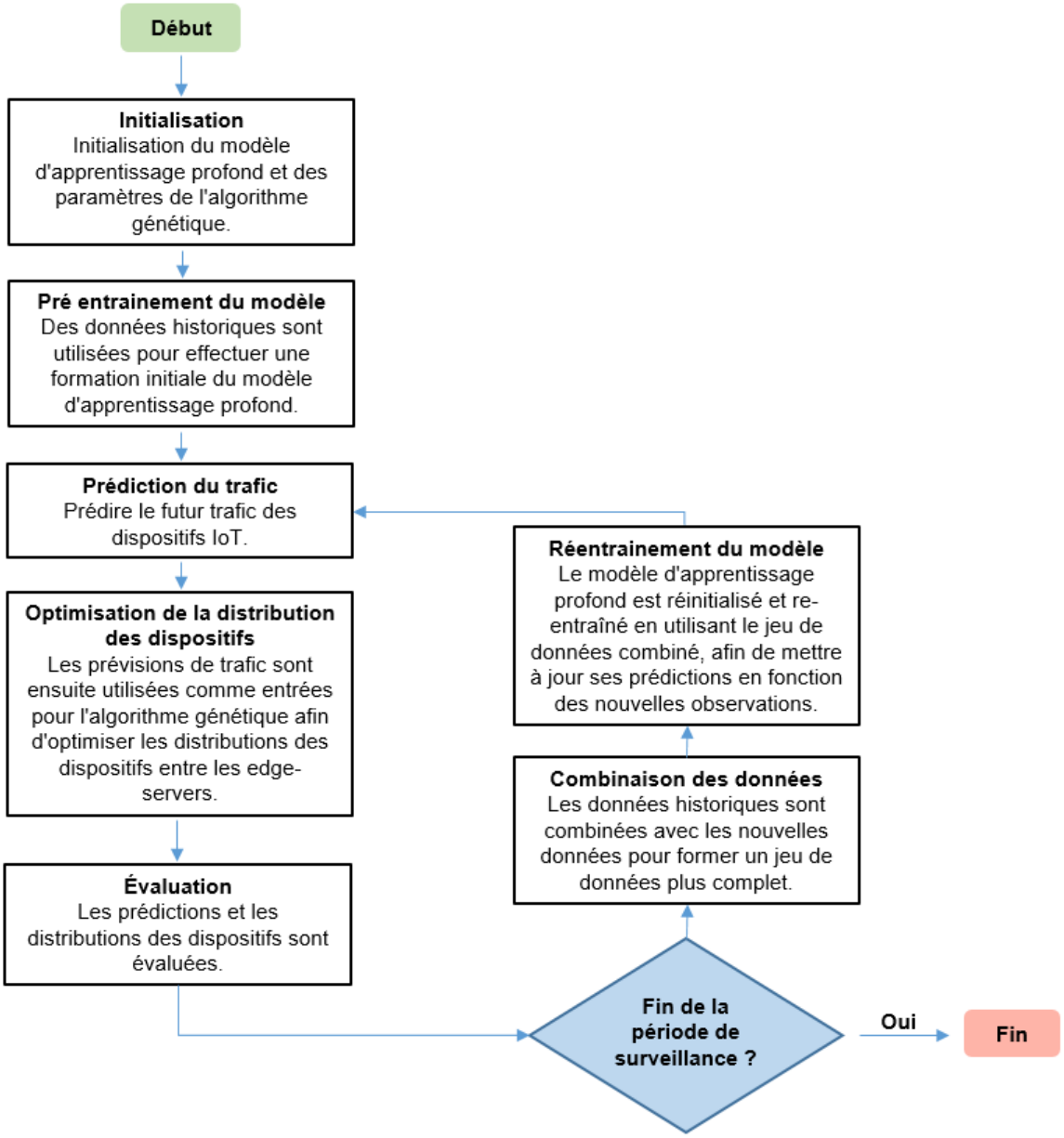


FIGURE 6.8 – Organigramme pour le clustering prédictif dynamique des réseaux IoT gérés par serveurs edge.

TABLE 6.1 – Paramètres d'entraînement du modèle LSTM.

Paramètre	**valeur**
Nombre d'époques d'apprentissage	100
Taille du lot	32
Optimiseur	adams
Fonction de perte	MSE
Fonction d'activation	ReLU

— Nombre d'époques d'apprentissage (epochs) : nombre de fois où le modèle parcourt l'ensemble de données d'entraînement.
— Taille du lot (batch size) : nombre d'échantillons utilisés dans chaque itération avant de mettre à jour les poids.
— Optimiseur (optimizer) : algorithme pour ajuster les poids du modèle pendant l'apprentissage.
— Fonction de perte (loss function) : mesure de l'erreur entre les prédictions du modèle et les valeurs réelles.
— Fonction d'activation (activation function) : fonction qui introduit la non-linéarité dans le modèle, facilitant l'apprentissage de relations complexes.

Les paramètres de l'algorithme génétique sont résumés dans le tableau 6.2.

TABLE 6.2 – Paramètres de l'algorithme génétique.

Paramètre	valeur
Nombre de générations	100
Taux de mutation	10%

Nous évaluons la précision du modèle de prédiction en utilisant l'erreur quadratique moyenne (Mean Squared Error, MSE), l'erreur absolue moyenne (Mean Absolute Error, MAE) et l'erreur quadratique moyenne (Root Mean Squared Error, RMSE) [188]. Nous comparons notre modèle de prédiction du trafic avec celui d'Abdellah *et al.* [189], et nous évaluons l'efficacité opérationnelle et le temps des deux algorithmes. Par la suite, notre analyse s'étend à l'évaluation de l'équilibrage de la charge entre les serveurs edge. Nous comparons les performances de notre proposition avec celles d'une approche de clustering statique. Nous évaluons la répartition de la charge entre les serveurs dans chaque scénario, en soulignant les variations et les ajustements induits par le clustering dynamique. En outre, nous mesurons l'écart-type entre les charges des serveurs edge

pour chaque configuration de clustering, ce qui donne un aperçu détaillé de la stabilité et de la répartition homogène de la charge dans ces deux contextes.

6.5.1 Évaluation des prédictions

La figure 6.9 illustre la phase de pré-entraînement de notre modèle. Les figures 6.10, 6.11 et 6.12 illustrent les prédictions de notre modèle à différents stades d'entraînement.

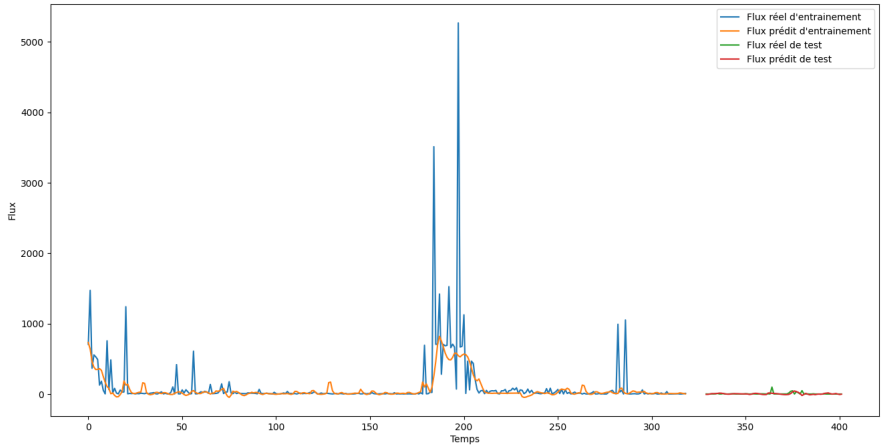


FIGURE 6.9 – Phase de préentraînement.

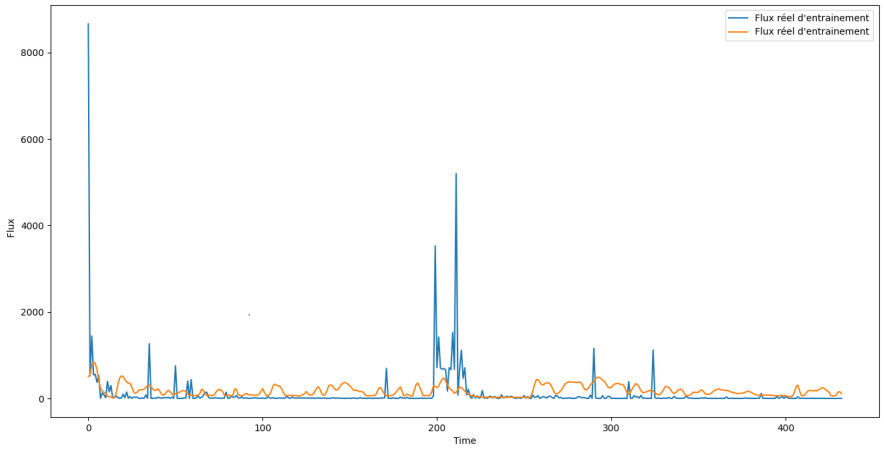


FIGURE 6.10 – Première phase de prédiction.

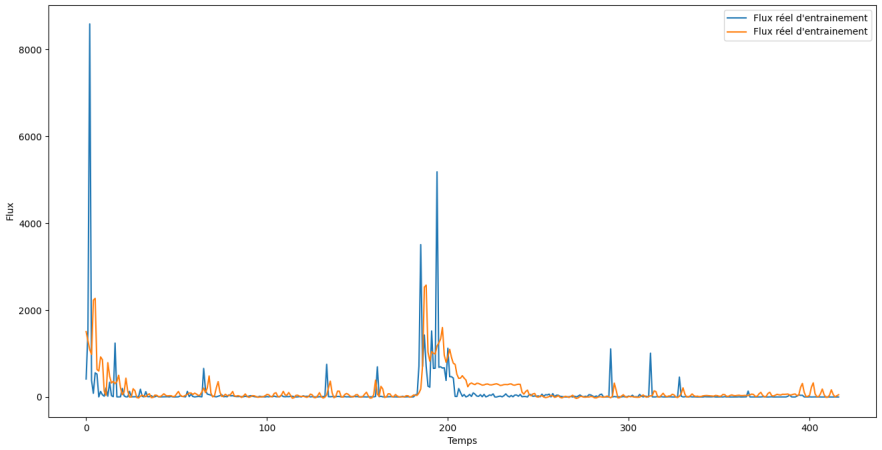


FIGURE 6.11 – Deuxième phase de prédiction.

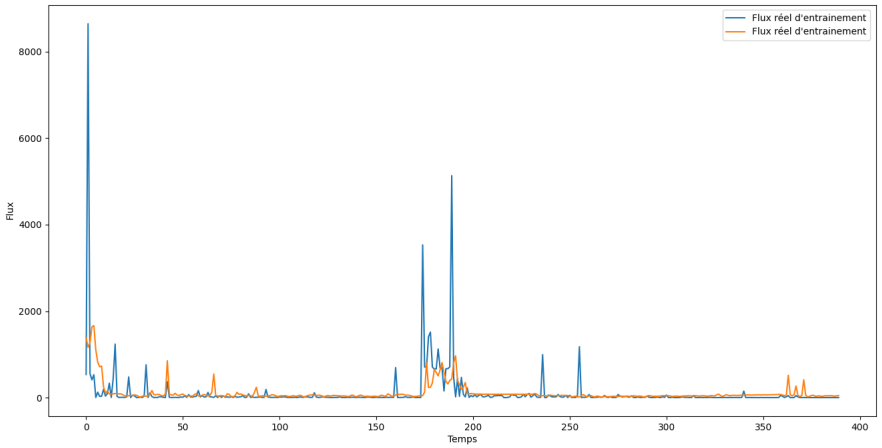


FIGURE 6.12 – Troisième phase de prédiction.

La figure 6.9 montre les prédictions du modèle basées uniquement sur les données historiques de la phase de pré-entraînement, tandis que les figures 6.10, 6.11 et 6.12 intègrent les données des phases ultérieures du processus d'apprentissage. La comparaison montre que le modèle offre des prédictions solides lorsqu'il est formé sur des données historiques. L'intégration de données issues des phases ultérieures permet également d'obtenir de bonnes prédictions, ce qui suggère la capacité du modèle à généraliser et à anticiper des tendances plus complexes dans le flux du réseau IoT au fil du temps. Cette évolution souligne l'efficacité du modèle dans la gestion de la dynamique évolutive des données IoT au cours des différentes phases.

6.5.2 Évaluation de la précision

Le tableau 6.3 présente une évaluation comparative des performances de prédiction des phases d'apprentissage par rapport au modèle de prédiction d'Abdellah *et al.*

Phase	**Modèle proposé**			**Abdellah et al.**		
	MSE	**MAE**	**RMSE**	**MSE**	**MAE**	**RMSE**
Première prédiction	3.92×10^{-5}	0.0027	0.0063	1.09×10^{-4}	0.0050	0.0104
Deuxième prédiction	2.33×10^{-4}	0.0066	0.0153	4.16×10^{-4}	0.0067	0.0204
Troisième prédiction	6.78×10^{-5}	0.0031	0.0082	1.45×10^{-4}	0.0039	0.0120

TABLE 6.3 – Évaluation de la précision du modèle avec MSE, MAE et RMSE pour le modèle proposé vs celui d'Abdellah *et al.*

Des résultats variables de MSE, MAE et RMSE sont notés. La MSE est faible pour les deux modèles au cours des premières phases, ce qui indique un accord très étroit entre les prédictions du modèle et les valeurs réelles. De même, la MAE et la RMSE sont également faibles, ce qui souligne la grande précision de la modélisation des données historiques au cours de cette phase initiale. En revanche, la MAE augmente légèrement pour les autres phases, en particulier la deuxième phase, ce qui indique une variation accrue entre les prédictions du modèle et les données réelles ; de même, les autres mesures augmentent. Ces résultats peuvent être attribués à des changements dans la dynamique des données au fil du temps.

Notre modèle présente des performances supérieures à celles du modèle d'Abdellah *et al.*, avec une MSE, un MAE et un RMSE inférieurs dans les phases de prédiction initiale, de deuxième et de troisième prédiction. Ces résultats démontrent la précision et la fiabilité accrues de notre modèle par rapport au modèle de référence. Ce qui peut être expliqué par l'architecture plus avancée utilisée dans notre modèle. Notre architecture comprend trois couches LSTM ; en outre, l'utilisation de dropouts et de fonctions d'activation améliore la robustesse du modèle, en réduisant le surapprentissage et en encourageant l'apprentissage de représentations plus stables et plus générales. Cette combinaison stratégique garantit non seulement des performances précises et cohérentes dans des conditions variables, mais améliore également la capacité du modèle à maintenir des prédictions fiables dans des environnements réels et dynamiques.

6.5.3 Efficacité opérationnelle et temps

Le tableau 6.4 résume les temps d'entraînement et de prédiction des modèles comparés.

TABLE 6.4 – Évaluation de l'efficacité opérationnelle et de la durée du modèle de prédiction proposé vs celui d'Abdellah *et al.*

Modèle	Phase	Temps d'entraînement (secondes)	Temps de prédiction (secondes)	Score d'efficacité
Modèle proposé	Préentraînement	357.9777674674988	-	
	Première phase	512.941353559494	1.5332057476043701	Élevé
	Deuxième phase	748.9963972568512	1.348384380340576 2	
	Troisième phase	-	1.274889230728149 4	
Modèle d'Abdellah *et al.*	Préentraînement	126.138887882232 67	-	
	Première phase	281.3424108028412	1.4414212703704834	Modéré
	Deuxième phase	174.0305891036987 3	1.4520628452301025	
	Troisième phase	-	0.4066526889801025 4	

Le modèle proposé présente des temps d'entraînement nettement plus longs que le modèle de référence. Cela s'explique probablement par le fait que le modèle proposé est plus complexe et qu'il utilise des techniques d'apprentissage plus approfondies pour améliorer la précision. Toutefois, le modèle proposé présente des performances acceptables en termes de temps de prédiction, ce qui est avantageux pour les applications nécessitant une réactivité rapide.

Malgré l'augmentation du temps d'entraînement, l'efficacité de prédiction du modèle proposé suggère un compromis entre la durée de la formation et la précision de la prédiction. L'efficacité opérationnelle du modèle proposé peut être considérée comme acceptable, compte tenu de ses performances supérieures en termes de métriques d'erreur pendant les phases clés de la prédiction.

En résumé, bien que le modèle proposé prenne plus de temps en termes d'entraînement, il compense cela par des temps de prédiction plus rapides, ce qui pourrait être un facteur décisif en fonction des exigences spécifiques de l'application.

6.5.4 Évaluation de la charge des serveurs edge

La figure 6.13 montre les distributions de charge sur les serveurs edge dans notre proposition, une approche statique et Abdellah *et al.* combiné avec notre allocation dynamique par GAs. La principale observation est que les approches dynamiques offrent des charges de serveurs plus équitables que l'approche statique.

La figure 6.14 illustre l'écart type des répartitions de charge entre les différentes approches. L'écart-type vient appuyer les résultats précédents. La répartition dynamique de la charge présente généralement un écart type de la charge entre les serveurs inférieur à celui de l'approche statique. Cela peut s'expliquer par le fait que les approches dynamiques peuvent s'adapter aux variations du trafic plus efficacement que les approches statiques, car elles prennent en compte les prévisions du trafic et ajustent l'équilibrage de la charge en conséquence. En outre, le clustering de machines est basé sur GAs. Les GAs sont particulièrement efficaces pour résoudre des problèmes d'optimisation complexes. L'algorithme génétique utilise ses mécanismes évolutifs pour de minimiser l'écart type entre les charges des serveurs, permettant ainsi une bonne répartition de la charge.

Malgré les variations entre les flux prédits et réels illustrées dans les figures 6.9, 6.10, 6.11, et 6.12, notre proposition améliore avec succès la répartition de la charge entre les serveurs edge. Les résultats notés par notre approche sont meilleurs par rapport à Abdellah et al. avec allocation dynamique. Comme nous l'avons vu précédemment, notre modèle est plus précis que celui d'Abdellah *et al.*, par conséquent l'écart type de

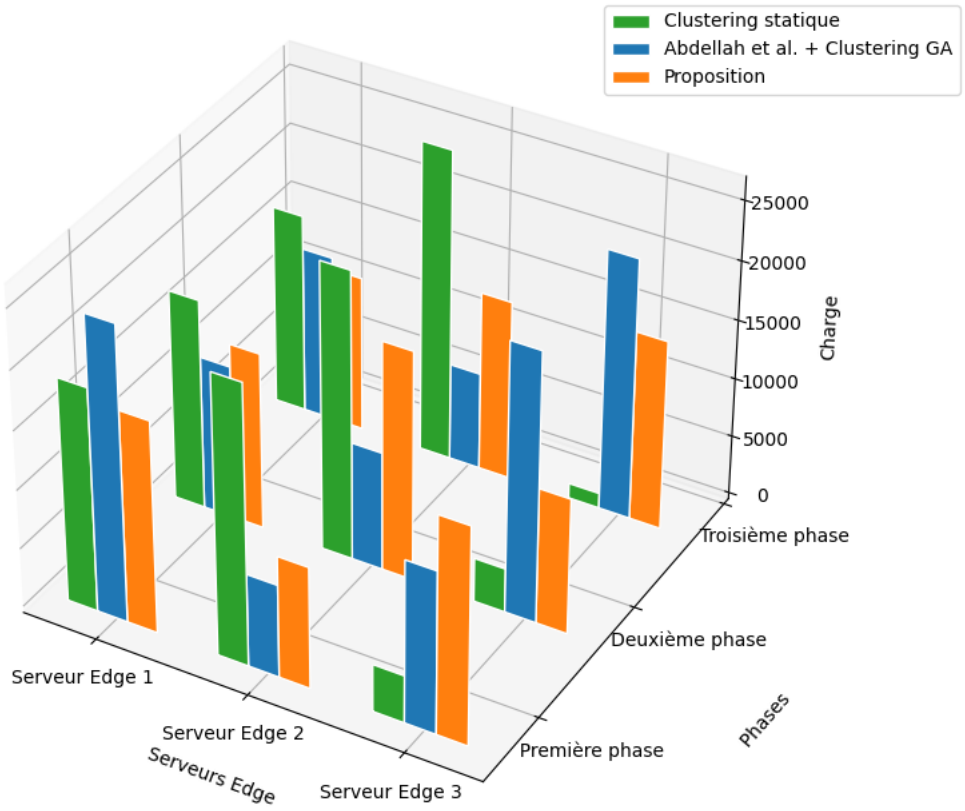


FIGURE 6.13 – Évaluation de la charge des serveurs edge de notre proposition vs le modèle prédictif d'Abdellah *et al.* avec notre allocation dynamique par algorithmes génétiques vs une approche statique.

la charge entre les serveurs, mesuré par notre proposition, est plus proche de la charge réelle, donc il est plus faible et meilleur.

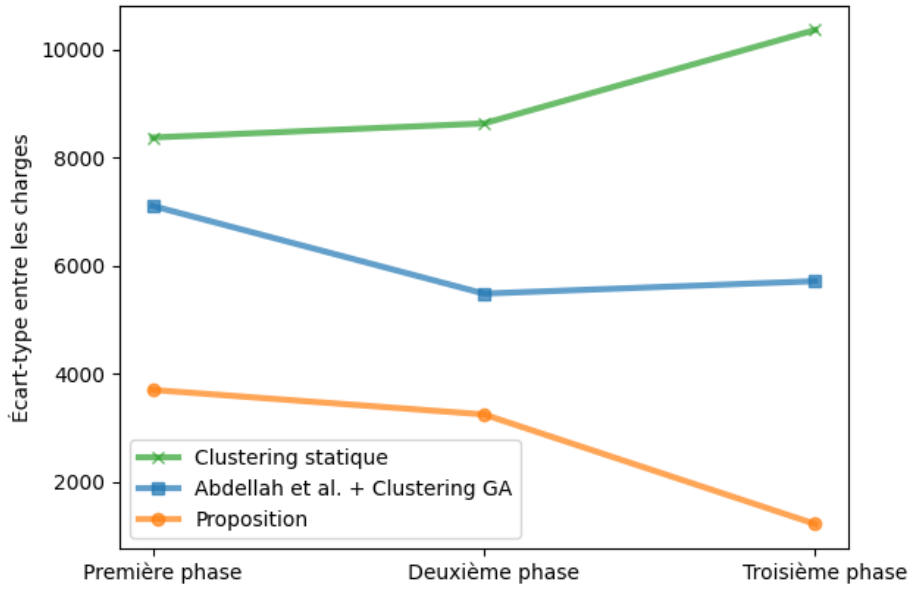


FIGURE 6.14 – Écart type de la charge entre les serveurs dans notre proposition vs le modèle prédictif d'Abdellah *et al.* avec notre allocation dynamique par algorithmes génétiques vs une approche statique.

6.6 Conclusion

Ce chapitre a mis en évidence l'efficacité des techniques d'apprentissage automatique supervisé, en particulier le deep learning, pour optimiser l'équilibrage de la charge entre les serveurs dans les réseaux IoT gérés par serveurs edge. En utilisant des réseaux de neurones à mémoire à long terme, nous avons pu analyser les données historiques et prédire les variations de trafic avec une précision accrue, tout en intégrant un algorithme génétique pour assurer une répartition optimale de la charge, ce qui permet d'améliorer les performances des systèmes IoT gérés par serveurs edge.

Les résultats de la simulation reflètent l'efficacité de notre approche. Le modèle prédictif anticipe avec succès les fluctuations de la charge des appareils, ce qui permet une adaptation en temps réel pour optimiser la répartition des machines entre les serveurs périphériques. D'autre part, l'algorithme génétique équilibre la charge entre les serveurs de manière optimale.

Divers aspects peuvent être développés davantage dans les travaux futurs. Le modèle de prédiction peut être raffiné. Des techniques d'apprentissage profond plus avancées peuvent être explorées pour améliorer la précision de nos prédictions de trafic IoT. De plus, le modèle d'apprentissage profond repose sur un ensemble d'hyperparamètres qui doivent être ajustés pour fournir de bons résultats. La performance du modèle peut être

améliorée en testant une plus grande variété de combinaisons d'hyperparamètres ou en utilisant des techniques d'optimisation d'hyperparamètres.

Dans notre travail, les données historiques sont combinées à de nouvelles données pour réentraîner le modèle. Cependant, il est essentiel d'être prudent lors de la combinaison de données historiques et récentes, en particulier si le contexte du problème change au fil du temps. Si les anciennes données deviennent obsolètes ou ne sont plus représentatives de la situation actuelle, cela pourrait entraîner un surajustement des données obsolètes. Des mises à jour progressives et une surveillance continue peuvent être mises en œuvre pour améliorer les performances du modèle.

Dans notre travail, nous avons pris en compte le nombre de flux pour l'équilibrage de la charge. D'autres facteurs, tels que le protocole et les numéros de port, peuvent être pris en compte lors de la prise de décisions relatives à l'équilibrage de la charge afin d'améliorer les performances.

Dans notre clustering prédictif dynamique, le modèle d'apprentissage profond est exécuté à partir d'un serveur central. Le concept d'apprentissage profond distribué, dans lequel les tâches de calcul sont réparties sur plusieurs serveurs ou appareils, peut être exploité pour permettre une gestion parallèle et une amélioration potentielle de la réactivité du système et des performances à grande échelle.

Conclusion générale et perspectives

Dans cette thèse, nous avons exploré et développé des solutions innovantes pour relever les défis complexes liés à l'organisation et à l'optimisation des réseaux de l'IoT. À travers trois contributions majeures, nous avons démontré l'efficacité de l'application de diverses techniques de machine learning pour améliorer les performances, l'efficacité énergétique et la robustesse des réseaux IoT.

Grâce à la thèse et à ses contributions, trois problèmes de clustering ont été abordés : le problème du clustering des dispositifs, la sélection des CHs et l'équilibrage de la charge dans les réseaux IoT gérés par edge computing. Nos contributions ont chacune exploité une approche distincte du machine learning pour répondre à ces défis.

Notre première contribution a abordé le problème de clustering des nœuds dans les réseaux de capteurs sans fil en utilisant une approche non supervisée. L'extension de l'algorithme U-k-means, couplée à une méthode de sélection des CHs basée sur des algorithmes génétiques, a permis une organisation plus efficace et autonome des WSNs, sans nécessiter la spécification préalable du nombre de clusters. La deuxième contribution s'est concentrée sur le problème de la sélection dynamique des CHs en utilisant une approche par renforcement. Notre méthode basée sur le Q-learning, avec une fonction de récompense multi-objectifs, a démontré une amélioration significative dans l'optimisation du routage, l'équilibrage de la charge, et l'efficacité énergétique des réseaux IoT. Enfin, notre troisième contribution a traité le problème de l'équilibrage de la charge dans les réseaux IoT gérés par des edge serveurs, en utilisant une approche supervisée. La combinaison des réseaux LSTM pour la prédiction du trafic et des algorithmes génétiques pour la répartition dynamique de la charge a permis d'améliorer considérablement la surveillance du trafic et l'efficacité globale du système.

Les travaux présentés dans cette thèse ouvrent la voie à plusieurs pistes de recherche futures :

— Explorer des approches hybrides combinant plusieurs techniques de ML afin de tirer parti des forces de chaque paradigme.

— Les techniques de ML supervisé ne peuvent être utilisées que lorsque des données d'entraînement sont disponibles. La fiabilité et la disponibilité de ces données sont des critères cruciaux pour ce type d'algorithme. Il existe un manque d'ensembles de données d'entraînement spécifiques pour le clustering des réseaux IoT. La collecte et la construction de tels ensembles sont nécessaires pour faciliter l'exploitation de ces algorithmes.
— Les techniques de ML offrent plusieurs avantages par rapport aux approches traditionnelles pour réaliser un clustering efficace dans les réseaux IoT. Toutefois, les modèles de ML sont exposés à diverses menaces et vulnérabilités en matière de sécurité. Par conséquent, la mise en place de mécanismes de sécurisation des techniques de ML, notamment lors des phases d'entraînement et de validation des ensembles de données, doit être envisagée.
— L'exploitation de l'apprentissage fédéré pourrait être explorée afin de décentraliser davantage le processus de décision.
— Compte tenu de la complexité et du coût de la plupart des algorithmes de ML, leur déploiement distribué dans les systèmes IoT peut imposer certaines contraintes. Ces réseaux étant généralement constitués de dispositifs aux ressources limitées, la réalisation de la phase d'apprentissage en mode hors ligne pourrait être étudiée pour surmonter ce défi.

Bibliographie

[1] Malha Merah, Zibouda Aliouat, and Hakim Mabed. Dynamic load balancing of traffic in the iot edge computing environment using a clustering approach based on deep learning and genetic algorithms. *Cluster Computing*, 2024.

[2] Malha Merah, Zibouda Aliouat, and Hakim Mabed. Cross-joint k-means and genetic scheme for internet of things sensor clustering. *International Journal of Sensor Networks*, 45(1) :1–15, 2024.

[3] Malha Merah, Zibouda Aliouat, Yasmine Harbi, and Mohamed Sofiane Batta. Machine learning-based clustering protocols for internet of things networks : An overview. *International Journal of Communication Systems*, page e5487, 2023.

[4] Malha Merah, Zibouda Aliouat, and Hakim Mabed. A novel cluster head selection algorithm based on q-learning for internet of things networks. In *2024 International Conference on Telecommunications and Intelligent Systems (ICTIS)*. IEEE, 2024.

[5] Malha Merah and Zibouda Aliouat. Unsupervised k-means for energy conservation in iot networks. In *2022 First International Conference on Computer Communications and Intelligent Systems (I3CIS)*, pages 61–66. IEEE, 2022.

[6] Malha Merah, Zibouda Aliouat, and Mohamed Sofiane Batta. A hybrid neural network and graph theory based clustering protocol for dynamic iot networks. In *2022 International Conference on Advanced Aspects of Software Engineering (ICAASE)*, pages 1–7. IEEE, 2022.

[7] Malha Merah, Zibouda Aliouat, and Chafia Kara-Mohamed. An energy efficient self organizing map based clustering protocol for iot networks. In *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages 197–203. IEEE, 2022.

[8] Mohamed Sofiane Batta, Zibouda Aliouat, Hakim Mabed, and Malha Merah. An improved lifetime optimization clustering using kruskal's mst and batteries aging

for iot networks. In *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2022.

[9] Mohamed Sofiane Batta, Hakim Mabed, Zibouda Aliouat, and Saad Harous. A distributed multi-hop intra-clustering approach based on neighbors two-hop connectivity for iot networks. *Sensors*, 21(3) :873, 2021.

[10] Jyoti Maini and Shalli Rani. Introduction to iot and wsn. In *WSN and IoT*, pages 1–34. CRC Press, 2024.

[11] Najmul Hassan, Saira Gillani, Ejaz Ahmed, Ibrar Yaqoob, and Muhammad Imran. The role of edge computing in internet of things. *IEEE communications magazine*, 56(11) :110–115, 2018.

[12] Georgios Lampropoulos, Kerstin Siakas, and Theofylaktos Anastasiadis. Internet of things in the context of industry 4.0 : An overview. *International Journal of Entrepreneurial Knowledge*, 7(1), 2019.

[13] Ruby Dwivedi, Divya Mehrotra, and Shaleen Chandra. Potential of internet of medical things (iomt) applications in building a smart healthcare system : A systematic review. *Journal of oral biology and craniofacial research*, 12(2) :302–318, 2022.

[14] Sara Oleiro Araújo, Ricardo Silva Peres, José Barata, Fernando Lidon, and José Cochicho Ramalho. Characterising the agriculture 4.0 landscape—emerging trends, challenges and opportunities. *Agronomy*, 11(4) :667, 2021.

[15] Moussa Aboubakar, Mounir Kellil, and Pierre Roux. A review of iot network management : Current status and perspectives. *Journal of King Saud University-Computer and Information Sciences*, 34(7) :4163–4176, 2022.

[16] Waleed Ejaz, Muhammad Naeem, Adnan Shahid, Alagan Anpalagan, and Minho Jo. Efficient energy management for the internet of things in smart cities. *IEEE Communications magazine*, 55(1) :84–91, 2017.

[17] Anit Chaudhary. Internet of things (iot) : Research challenges and future applications. *International Journal of Emerging Trends in Science and Technology*, 2022.

[18] J. Amutha,S. Sharma, and S. K. Sharma. Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine

learning techniques : Review, taxonomy, research findings, challenges and future directions. *Computer Science Review*, 40 :100376, 2021.

[19] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani. Deep learning for iot big data and streaming analytics : A survey. *IEEE Communications Surveys & Tutorials*, 20(4) :2923–2960, 2018.

[20] Mohamed Said Frikha, Sonia Mettali Gammar, Abdelkader Lahmadi, and Laurent Andrey. Reinforcement and deep reinforcement learning for wireless internet of things : A survey. *Computer Communications*, 2021.

[21] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Understanding the internet of things : definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56 :122–140, 2017.

[22] Oracle. What is iot ? `https://www.oracle.com/internet-of-things/what-is-iot/`, sans date. Consulté le 05/06/2024.

[23] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE, 2000.

[24] Gartner. Internet of things (iot). `https://www.gartner.com/en/information-technology/glossary/internet-of-things`, sans date. Consulté le 05/06/2024.

[25] Mahdi H Miraz, Maaruf Ali, Peter S Excell, and Rich Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). *2015 Internet Technologies and Applications (ITA)*, pages 219–224, 2015.

[26] Red Hat. L'iot (internet des objets), qu'est-ce que c'est ? `https://www.redhat.com/fr/topics/internet-of-things/what-is-iot`, sans date. Consulté le 20/09/2024.

[27] Yasmine Harbi, Zibouda Aliouat, Allaoua Refoufi, and Saad Harous. Recent security trends in internet of things : A comprehensive survey. *IEEE Access*, 9 :113292–113314, 2021.

[28] Alfred Heller. The sensing internet—a discussion on its impact on rural areas. *Future Internet*, 7(4) :363–371, 2015.

[29] Ana Carolina Borges Monteiro, Reinaldo Padilha França, Rangel Arthur, and Yuzo Iano. An overview of the internet of medical things (iomt) : Applications, benefits, and challenges. *Security and Privacy Issues in Internet of Medical Things*, pages 83–98, 2023.

[30] Baofeng Ji, Xueru Zhang, Shahid Mumtaz, Congzheng Han, Chunguo Li, Hong Wen, and Dan Wang. Survey on the internet of vehicles : Network architectures and applications. *IEEE Communications Standards Magazine*, 4(1) :34–41, 2020.

[31] Vasiliki Demertzi, Stavros Demertzis, and Konstantinos Demertzis. An overview of privacy dimensions on the industrial internet of things (iiot). *Algorithms*, 16(8) :378, 2023.

[32] Seyed Mahdi Jameii, Romina Sadat Zamirnaddafi, and Reza Rezabakhsh. Internet of flying things security : A systematic review. *Concurrency and Computation : Practice and Experience*, 34(24) :e7213, 2022.

[33] Abdullah Alabdulatif, Navod Neranjan Thilakarathne, Zaharaddeen Karami Lawal, Khairul Eahsun Fahim, and Rufai Yusuf Zakari. Internet of nano-things (iont) : A comprehensive review from architecture to security and privacy challenges. *Sensors*, 23(5) :2807, 2023.

[34] Jibran Saleem, Mohammad Hammoudeh, Umar Raza, Bamidele Adebisi, and Ruth Ande. Iot standardisation : Challenges, perspectives and solution. In *Proceedings of the 2nd international conference on future networks and distributed systems*, pages 1–9, 2018.

[35] Mohamed Sofiane Batta, Hakim Mabed, Zibouda Aliouat, and Saad Harous. Battery state-of-health prediction-based clustering for lifetime optimization in iot networks. *IEEE Internet of Things Journal*, 10(1) :81–91, 2022.

[36] Ananda Mohon Ghosh and Katarina Grolinger. Edge-cloud computing for internet of things data analytics : Embedding intelligence in the edge with deep learning. *IEEE Transactions on Industrial Informatics*, 17(3) :2191–2200, 2020.

[37] Haroon Rashid Hammood Al Dallal et al. Clustering protocols for energy efficiency analysis in wsns and the iot. *Problems of Information Society*, pages 18–24, 2024.

[38] Francesco Restuccia, Salvatore D'Oro, and Tommaso Melodia. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal*, 5(6) :4829–4842, 2018.

[39] Chirihane Gherbi, Zibouda Aliouat, and Mohamed Benmohammed. A survey on clustering routing protocols in wireless sensor networks. *Sensor Review*, 2017.

[40] Priyantha Kumarawadu, Dan J Dechene, Marco Luccini, and Allan Sauer. Algorithms for node clustering in wireless sensor networks : A survey. In *2008 4th International Conference on Information and Automation for Sustainability*, pages 295–300. IEEE, 2008.

[41] Amin Shahraki, Amir Taherkordi, Øystein Haugen, and Frank Eliassen. Clustering objectives in wireless sensor networks : A survey and research direction analysis. *Computer Networks*, 180 :107376, 2020.

[42] Dave Evans. The internet of things : How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011) :1–11, 2011.

[43] Sunil Kumar Singh, Prabhat Kumar, and Jyoti Prakash Singh. A survey on successors of leach protocol. *Ieee Access*, 5 :4298–4328, 2017.

[44] Abhilash Singh, Sandeep Sharma, and Jitendra Singh. Nature-inspired algorithms for wireless sensor networks : A comprehensive survey. *Computer Science Review*, 39 :100342, 2021.

[45] Tom M Mitchell. Machine learning. 1(9), 1997.

[46] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3) :210–229, 1959.

[47] Red Hat. L'apprentissage automatique, ou machine learning, qu'est-ce que c'est ? `https://www.redhat.com/fr/topics/ai/what-is-machine-learning`, sans date. Consulté le 02/10/2024.

[48] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10) :78–87, 2012.

[49] Ian Goodfellow. Deep learning, 2016.

[50] Zheng Zhou, Cheng Qiu, and Yufan Zhang. A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models. *Scientific Reports*, 13(1) :22420, 2023.

[51] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[52] Frank Emmert-Streib and Matthias Dehmer. Taxonomy of machine learning paradigms : A data-centric perspective. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 12(5) :e1470, 2022.

[53] Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain. Machine learning in iot security : Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, 22(3) :1686–1721, 2020.

[54] Hanuman Verma, Akshansh Gupta, Jyoti Singh Kirar, Mukesh Prasad, and CT Lin. Introduction to computational methods : Machine and deep learning perspective. In *Computational Intelligence Aided Systems for Healthcare Domain*, pages 1–32. CRC Press, 2023.

[55] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. Fundamentals of artificial neural networks and deep learning. In *Multivariate statistical machine learning methods for genomic prediction*, pages 379–425. Springer, 2022.

[56] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE access*, 7 :53040–53065, 2019.

[57] Zhice Fang, Yi Wang, Ling Peng, and Haoyuan Hong. Integration of convolutional neural network and conventional machine learning classifiers for landslide susceptibility mapping. *Computers & Geosciences*, 139 :104470, 2020.

[58] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning : A brief survey. *IEEE Signal Processing Magazine*, 34(6) :26–38, 2017.

[59] Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms : State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110 :104743, 2022.

[60] Anil Khandelwal and Yogendra Kumar Jain. An efficient k-means algorithm for the cluster head selection based on saw and wpm. *International Journal of Advanced Computer Research*, 8(37) :191–202, 2018.

[61] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm : A comprehensive survey and performance evaluation. *Electronics*, 9(8) :1295, 2020.

[62] Garima Sharma, Praveen Kumar, and Laxmi Shrivastava. An efficient performance of enhanced bellman-ford algorithm in wireless sensor network using k-medoid clustering. In *International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering & Technology*, pages 52–65. Springer, 2019.

[63] Abdelrahman Radwan, Nazhatul Hafizah Kamarudin, Mahmud Iwan Solihin, CK Ang, H Leong, and M Rizon. Implementation of the x-means clustering algorithm for wireless sensor networks. In *International Conference on Artificial Life and Robotics (ICAROB 2020)[Internet]. ICAROB2020*, pages 333–7, 2020.

[64] Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion : background, derivation, and applications. *Wiley Interdisciplinary Reviews : Computational Statistics*, 4(2) :199–203, 2012.

[65] Parag Kumar Guha Thakurta and Soumali Roy. A decentralized fuzzy c-means minimal clustering protocol for energy efficient wireless sensor network. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 24–29. IEEE, 2018.

[66] C Challoumis. Fuzzy logic concepts and the qe (quantification of every-thing) method in economics. *Web of Scholars : Multidimensional Research Journal*, 3(4) :1–25, 2024.

[67] M. Merah, Z. Aliouat, and C. Kara-mohamed. An energy efficient self organizing map based clustering protocol for iot networks. In *IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2022.

[68] Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1) :31–39, 2017.

[69] Abdollah Amirkhani, Hosna Nasiriyan-Rad, and Elpiniki I Papageorgiou. A novel fuzzy inference approach : neuro-fuzzy cognitive map. *International Journal of Fuzzy Systems*, 22 :859–872, 2020.

[70] Jesse Clifton and Eric Laber. Q-learning : Theory and applications. *Annual Review of Statistics and Its Application*, 7(1) :279–301, 2020.

[71] Lionel Jouffe. Fuzzy inference system learning by reinforcement methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3) :338–355, 1998.

[72] Melrose Roderick, James MacGlashan, and Stefanie Tellex. Implementing the deep q-network. *arXiv preprint arXiv :1711.07478*, 2017.

[73] Andrew Brim. Deep reinforcement learning pairs trading with a double deep q-network. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0222–0227. IEEE, 2020.

[74] Chien-Liang Liu, Chuan-Chin Chang, and Chun-Jan Tseng. Actor-critic deep reinforcement learning for solving job shop scheduling problems. *Ieee Access*, 8 :71752–71762, 2020.

[75] Hadrien T Gayap and Moulay A Akhloufi. Deep machine learning for medical diagnosis, application to lung cancer detection : a review. *BioMedInformatics*, 4(1) :236–284, 2024.

[76] Dennis Herhausen, Stefan F Bernritter, Eric WT Ngai, Ajay Kumar, and Dursun Delen. Machine learning in marketing : Recent progress and future research directions. *Journal of Business Research*, 170 :114254, 2024.

[77] Hanyao Gao, Gang Kou, Haiming Liang, Hengjie Zhang, Xiangrui Chao, Cong-Cong Li, and Yucheng Dong. Machine learning in business and finance : a literature review and research opportunities. *Financial Innovation*, 10(1) :86, 2024.

[78] Tej Bahadur Shahi, Cheng Yuan Xu, Arjun Neupane, and William Guo. Machine learning methods for precision agriculture with uav imagery : a review. *Electronic Research Archive*, 30(12) :4277–4317, 2022.

[79] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications : a comprehensive survey. *Artificial Intelligence Review*, 55(2) :945–990, 2022.

[80] Abida Sharif, Jian Ping Li, Muhammad Asim Saleem, Gunasekaran Manogran, Seifedine Kadry, Abdul Basit, and Muhammad Attique Khan. A dynamic clustering technique based on deep reinforcement learning for internet of vehicles. *Journal of Intelligent Manufacturing*, 32(3) :757–768, 2021.

[81] Qingzhi Liu, Tiancong Xia, Long Cheng, Merijn Van Eijk, Tanir Ozcelebi, and Ying Mao. Deep reinforcement learning for load-balancing aware network control in iot edge systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(6) :1491–1502, 2021.

[82] Qingzhi Liu, Long Cheng, Tanir Ozcelebi, John Murphy, and Johan Lukkien. Deep reinforcement learning for iot network dynamic clustering in edge computing. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 600–603. IEEE, 2019.

[83] K Thangaramya, Kanagasabai Kulothungan, R Logambigai, M Selvi, Sannasi Ganapathy, and Arputharaj Kannan. Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in iot. *Computer Networks*, 151 :211–223, 2019.

[84] Santosh Soni and Manish Shrivastava. Novel learning algorithms for efficient mobile sink data collection using reinforcement learning in wireless sensor network. *Wireless Communications and Mobile Computing*, 2018, 2018.

[85] Zibouda Aliouat, Sana Benbekhouche, Lina Aliouat, and Mohamed Rahmani. Dynamic clustering based on q-learning for load balancing in iot. In *2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, pages 203–208. IEEE, 2022.

[86] MJ Rhesa and S Revathi. Energy efficiency random forest classification based data mustering in wireless sensor networks. In *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, pages 1–7. IEEE, 2021.

[87] Padmalaya Nayak, GK Swetha, Priyanka Kaushal, and DG Padhan. Cluster formation algorithm in wsns to optimize the energy consumption using self-organizing map. In *IoT and Analytics for Sensor Networks*, pages 11–22. Springer, 2022.

[88] Sultan Alkhliwi. Energy efficient cluster based routing protocol with secure ids for iot assisted heterogeneous wsn. *Energy*, 11(11), 2020.

[89] Mohd Adnan, Tazeem Ahmad, and Tao Yang. Type-2 fuzzy logic based energy-efficient cluster head election for multi-hop wireless sensor networks. In *2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pages 32–38. IEEE, 2021.

[90] Sankar Sennan, Somula Ramasubbareddy, Sathiyabhama Balasubramaniyam, Anand Nayyar, Mohamed Abouhawwash, and Noha A Hikal. T2fl-pso : Type-2 fuzzy logic-based particle swarm optimization algorithm used to maximize the lifetime of internet of things. *IEEE Access*, 9 :63966–63979, 2021.

[91] Quan Wang, Deyu Lin, Pengfei Yang, and Zhiqiang Zhang. A fuzzy-logic based energy-efficient clustering algorithm for the wireless sensor networks. In *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2018.

[92] Payam Rahimi and Chrysostomos Chrysostomou. Improving the network lifetime and performance of wireless sensor networks for iot applications based on fuzzy logic. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 667–674. IEEE, 2019.

[93] Armin Mazinani, Sayyed Majid Mazinani, and Mostafa Mirzaie. Fmcr-ct : An energy-efficient fuzzy multi cluster-based routing with a constant threshold in wireless sensor network. *Alexandria Engineering Journal*, 58(1) :127–141, 2019.

[94] Mostafa Mirzaie and Sayyed Majid Mazinani. Machfl-ft : a fuzzy logic based energy-efficient protocol to cluster heterogeneous nodes in wireless sensor networks. *Wireless Networks*, 25(8) :4597–4609, 2019.

[95] Abdulmughni Hamzah, Mohammad Shurman, Omar Al-Jarrah, and Eyad Taqieddin. Energy-efficient fuzzy-logic-based clustering technique for hierarchical routing protocols in wireless sensor networks. *Sensors*, 19(3) :561, 2019.

[96] Anagha Rajput and Vinoth Babu Kumaravelu. Fuzzy logic–based distributed clustering protocol to improve energy efficiency and stability of wireless smart sensor networks for farmland monitoring systems. *International Journal of Communication Systems*, 33(4) :e4239, 2020.

[97] Anshu Kumar Dwivedi and Awadesh K Sharma. Neef : A novel energy efficient fuzzy logic based clustering protocol for wireless sensor network. *Scalable Computing : Practice and Experience*, 21(3) :555–568, 2020.

[98] Akshay Verma, Sunil Kumar, Prateek Raj Gautam, Tarique Rashid, and Arvind Kumar. Fuzzy logic based effective clustering of homogeneous wireless sensor networks for mobile sink. *IEEE Sensors Journal*, 20(10) :5615–5623, 2020.

[99] Swapna Ch and Vijayashree R Budyal. Expectation maximization and fuzzy logic based energy efficient data collection in wireless sensor networks with mobile elements. In *2020 7th international conference on signal processing and integrated networks (SPIN)*, pages 21–26. IEEE, 2020.

[100] Kalaivanan Karunanithy and Bhanumathi Velusamy. Cluster-tree based energy efficient data gathering protocol for industrial automation using wsns and iot. *Journal of Industrial Information Integration*, 19 :100156, 2020.

[101] Pankaj Kumar Mishra and Shashi Kant Verma. Eefmcp : energy efficient fuzzy logic-based multi-clustering protocol. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15, 2021.

[102] ST Sheriba and D Hevin Rajesh. Energy-efficient clustering protocol for wsn based on improved black widow optimization and fuzzy logic. *Telecommunication Systems*, 77(1) :213–230, 2021.

[103] Bharat Bhushan and Gadadhar Sahoo. Fleac : Fuzzy logic-based energy adequate clustering protocol for wireless sensor networks using improved grasshopper optimization algorithm. *Wireless Personal Communications*, pages 1–34, 2021.

[104] AR Rajeswari, Kanagasabai Kulothungan, Sannasi Ganapathy, and Arputharaj Kannan. Trusted energy aware cluster based routing using fuzzy logic for wsn in iot. *Journal of Intelligent & Fuzzy Systems*, (Preprint) :1–15, 2021.

[105] Munuswamy Selvi, SVN Santhosh Kumar, Sannasi Ganapathy, Ayyasamy Ayyanar, Harichandran Khanna Nehemiah, and Arputharaj Kannan. An energy efficient clustered gravitational and fuzzy based routing algorithm in wsns. *Wireless Personal Communications*, 116(1) :61–90, 2021.

[106] Mohd Adnan, Liu Yang, Tazeem Ahmad, and Yang Tao. An unequally clustered multi-hop routing protocol based on fuzzy logic for wireless sensor networks. *IEEE Access*, 9 :38531–38545, 2021.

[107] Slaheddine Chelbi, Habib Dhahri, Majed Abdouli, Claude Duvallet, and Rafik Bouaziz. A new hybrid routing protocol for wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 28(4) :247–257, 2018.

[108] Shengchao Su and Shuguang Zhao. An optimal clustering mechanism based on fuzzy-c means for wireless sensor networks. *Sustainable Computing : Informatics and Systems*, 18 :127–134, 2018.

[109] Quyuan Wang, Songtao Guo, Jianji Hu, and Yuanyuan Yang. Spectral partitioning and fuzzy c-means based clustering algorithm for big data wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2018(1) :1–11, 2018.

[110] Sanjoy Mondal, Saurav Ghosh, and Pratik Dutta. Energy efficient data gathering in wireless sensor networks using rough fuzzy c-means and aco. In *Industry Interactive Innovations in Science, Engineering and Technology*, pages 163–172. Springer, 2018.

[111] Abhilasha Jain and Ashok Kumar Goel. Energy efficient algorithm for wireless sensor network using fuzzy c-means clustering. *International (IJACSA) International Journal of Advanced Computer Science and Applications*, 2018.

[112] Runze Wan, Naixue Xiong, Qinghui Hu, Haijun Wang, and Jun Shang. Similarity-aware data aggregation using fuzzy c-means approach for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2019(1) :1–11, 2019.

[113] T Mamatha and P Aishwarya. An efficient cluster based routing protocol using hybrid fcm-q leach for vehicular ad hoc networks. *Int. J. Appl. Eng. Res*, 14(7) :1604–1612, 2019.

[114] Ahmed A Salman and Zainab T Alisa. Improving the network lifetime in wireless sensor network for internet of thing applications. *Al-Khwarizmi Engineering Journal*, 15(4) :79–90, 2019.

[115] C SriVenkateswaran and D Sivakumar. Secure cluster-based data aggregation in wireless sensor networks with aid of ecc. *International Journal of Business Information Systems*, 31(2) :153–169, 2019.

[116] Richa Sharma, Vasudha Vashisht, and Umang Singh. Eefcm-de : energy-efficient clustering based on fuzzy c means and differential evolution algorithm in wsns. *IET Communications*, 13(8) :996–1007, 2019.

[117] Anagha Rajput and Vinoth Babu Kumaravelu. Scalable and sustainable wireless sensor networks for agricultural application of internet of things using fuzzy c-means algorithm. *Sustainable Computing : Informatics and Systems*, 22 :62–74, 2019.

[118] Rahil Bensaid, Maymouna Ben Said, and Hatem Boujemaa. Fuzzy c-means based clustering algorithm in wsns for iot applications. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 126–130. IEEE, 2020.

[119] Ranjit Kumar, Sachin Tripathi, and Rajeev Agrawal. Enhanced shortest path routing protocol using fuzzy c-means clustering for compromised wsn to control risk. In *Design Frameworks for Wireless Networks*, pages 399–415. Springer, 2020.

[120] Susan Augustine and John Patrick Ananth. Taylor kernel fuzzy c-means clustering algorithm for trust and energy-aware cluster head selection in wireless sensor networks. *Wireless Networks*, 26(7) :5113–5132, 2020.

[121] Pedro Cumino, Kaled Maciel, Thaís Tavares, Helder Oliveira, Denis Rosário, and Eduardo Cerqueira. Cluster-based control plane messages management in software-defined flying ad-hoc network. *Sensors*, 20(1) :67, 2020.

[122] Ali Abdul-hussian Hassan, Wahidah Md Shah, Abdul-hussien Hassan Habeb, and Mohd Fairuz Iskandar Othman. Improved fuzzy c-means algorithm based on a novel mechanism for the formation of balanced clusters in wsns. *TELKOMNIKA*, 18(6) :2894–2902, 2020.

[123] Veervrat Singh Chandrawanshi, Rajiv Kumar Tripathi, and Rahul Pachauri. An intelligent low power consumption routing protocol to extend the lifetime of wireless sensor networks based on fuzzy c-means++ clustering algorithm. *Journal of Intelligent & Fuzzy Systems*, 38(5) :6561–6570, 2020.

[124] Nitesh Chouhan and SC Jain. An energy-efficient hybrid hierarchical clustering algorithm for wireless sensor devices in iot. In *International Conference on Advances in Computing and Data Sciences*, pages 1–14. Springer, 2021.

[125] Akhilesh Panchal and Rajat Kumar Singh. Ehcr-fcm : Energy efficient hierarchical clustering and routing using fuzzy c-means for wireless sensor networks. *Telecommunication Systems*, 76(2) :251–263, 2021.

[126] A Selva Reegan and V Kabila. Highly secured cluster based wsn using novel fcm and enhanced ecc-elgamal encryption in iot. *Wireless Personal Communications*, 118(2) :1313–1329, 2021.

[127] Djilali Moussaoui, Mourad Hadjila, Sidi Mohammed Hadj Irid, and Sihem Souiki. Clustered chain founded on ant colony optimization energy efficient routing scheme for under-water wireless sensor networks. *International Journal of Electrical & Computer Engineering (2088-8708)*, 11(6), 2021.

[128] Deepa Kalaimani, Zaheeruddin Zah, and Shruti Vashist. Energy-efficient density-based fuzzy c-means clustering in wsn for smart grids. *Australian Journal of Multi-Disciplinary Engineering*, 17(1) :23–38, 2021.

[129] D Karunkuzhali, B Meenakshi, and Keerthi Lingam. An adaptive fuzzy c means with seagull optimization algorithm for analysis of wsns in agricultural field with iot. 2021.

[130] Preeti Gupta, Sachin Tripathi, and Samayveer Singh. Energy efficient hotspot problem mitigation techniques using multiple mobile sink in heterogeneous wireless sensor network. *International Journal of Communication Systems*, 33(18) :e4641, 2020.

[131] Mohamed Elhoseny and K Shankar. Energy efficient optimal routing for communication in vanets via clustering model. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*, pages 1–14. Springer, 2020.

[132] Joy Iong Zong Chen and P Hengjinda. Enhanced dragonfly algorithm based k-medoid clustering model for vanet. *Journal of ISMAC*, 3(01) :50–59, 2021.

[133] Ramalingam Shanmugam and Baskaran Kaliaperumal. An energy-efficient clustering and cross-layer-based opportunistic routing protocol (corp) for wireless sensor network. *International Journal of Communication Systems*, 34(7) :e4752, 2021.

[134] B Iswarya and B Radha. Energy efficient clustering technique for vanet. In *Advances in Parallel Computing Technologies and Applications*, pages 105–113. IOS Press, 2021.

[135] Amine Faid, Mohamed Sadik, and Essaid Sabir. Eaca : An energy aware clustering algorithm for wireless iot sensors. In *2021 28th International Conference on Telecommunications (ICT)*, pages 1–6. IEEE, 2021.

[136] Amine Faid, Mohamed Sadik, and Essaid Sabir. Ihee : An improved hybrid energy efficient algorithm for wsn. In *Future of Information and Communication Conference*, pages 283–298. Springer, 2021.

[137] Abdelrahman Radwan, Nazhatul Hafizah Kamarudin, Mahmud Iwan Solihin, and Hungyang Leong. Slitting k-means clusters to x-means clusters for prolonging wireless sensor networks lifetime. In *AIP Conference Proceedings*, volume 2306, page 020008. AIP Publishing LLC, 2020.

[138] Abdelrahman Radwan, Nazhatul Kamarudin, Mahmud Iwan Solihin, Hungyang Leong, Mohamed Rizon, Desa Hazry, and Muhammad Azizi bin Azizan. X-means clustering for wireless sensor networks. *J. Robotics Netw. Artif. Life*, 7(2) :111–115, 2020.

[139] Abdelrahman Radwan, Nasser Abdellatif, Eyad Radwan, and Maryam Akhozahieh. Fitness function x-means for prolonging wireless sensor networks lifetime. *International Journal of Electrical and Computer Engineering*, 13(1) :465, 2023.

[140] Bindiya Jain, Gursewak Brar, and Jyoteesh Malhotra. Ekmt-k-means clustering algorithmic solution for low energy consumption for wireless sensor networks based on minimum mean distance from base station. In *Networking communication and data knowledge engineering*, pages 113–123. Springer, 2018.

[141] Mohamed Lehsaini and Meriem Bouchra Benmahdi. An improved k-means cluster-based routing scheme for wireless sensor networks. In *2018 International Symposium on Programming and Systems (ISPS)*, pages 1–6. IEEE, 2018.

[142] Nandoori Srikanth and MS Ganga Prasad. Efficient clustering protocol using fuzzy k-means and midpoint algorithm for lifetime improvement in wsns. *International Journal of Intelligent Engineering and Systems*, 11(4) :61–71, 2018.

[143] Lovepreet Kaur and Sandeep Kad. Modified eecpk-means mid-point algorithm for enhancing network life expactancy in wsn. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1262–1265. IEEE, 2018.

[144] Shashi Bhushan, Raju Pal, and Svetlana G Antoshchuk. Energy efficient clustering protocol for heterogeneous wireless sensor network : a hybrid approach using ga and k-means. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 381–385. IEEE, 2018.

[145] Madiha Razzaq, Devarani Devi Ningombam, and Seokjoo Shin. Energy efficient k-means clustering-based routing protocol for wsn using optimal packet size. In *2018 International Conference on Information Networking (ICOIN)*, pages 632–635. IEEE, 2018.

[146] GS Pavithra and NV Babu. Energy efficient hierarchical clustering using hacopso in wireless sensor networks. *International Journal of Innovative Technology and Exploring Engineering*, 8(12), 2019.

[147] Geetika Dhand and Shyam Sunder Tyagi. Smeer : secure multi-tier energy efficient routing protocol for hierarchical wireless sensor networks. *Wireless Personal Communications*, 105(1) :17–35, 2019.

[148] Xixi Dong, Yun Zhang, and Shujuan Yu. An uneven clustering routing protocol based on improved k-means algorithm for wireless sensor network in coal-mine. *J. Inf. Hiding Multim. Signal Process.*, 10(1) :53–62, 2019.

[149] Salim El Khediri, Walid Fakhet, Tarek Moulahi, Rehanullah Khan, Adel Thaljaoui, and Abdennaceur Kachouri. Improved node localization using k-means clustering for wireless sensor networks. *Computer Science Review*, 37 :100284, 2020.

[150] Vikas Srivastava, Sachin Tripathi, Karan Singh, and Le Hoang Son. Energy efficient optimized rate based congestion control routing in wireless sensor network. *Journal of Ambient Intelligence and Humanized Computing*, 11(3) :1325–1338, 2020.

[151] Botao Zhu, Ebrahim Bedeer, Ha H Nguyen, Robert Barton, and Jerome Henry. Improved soft-k-means clustering algorithm for balancing energy consumption in wireless sensor networks. *IEEE Internet of Things Journal*, 8(6) :4868–4881, 2020.

[152] Jay Kumar Jain. A coherent approach for dynamic cluster-based routing and coverage hole detection and recovery in bi-layered wsn-iot. *Wireless Personal Communications*, 114(1) :519–543, 2020.

[153] Kenechi G Omeke, Michael S Mollel, Metin Ozturk, Shuja Ansari, Lei Zhang, Qammer H Abbasi, and Muhammad Ali Imran. Dekcs : A dynamic clustering protocol to prolong underwater sensor networks. *IEEE Sensors Journal*, 21(7) :9457–9464, 2021.

[154] Amin Rezaeipanah, Parvin Amiri, Hamed Nazari, Musa Mojarad, and Hamid Parvin. An energy-aware hybrid approach for wireless sensor networks using re-clustering-based multi-hop routing. *Wireless Personal Communications*, 120(4) :3293–3314, 2021.

[155] Manuel Heusner, Thomas Keller, and Malte Helmert. Best-case and worst-case behavior of greedy best-first search. International Joint Conferences on Artificial Intelligence, 2018.

[156] Hossein Hajipour, Hamed Behzadi Khormuji, and Habib Rostami. Odma : a novel swarm-evolutionary metaheuristic optimizer inspired by open source development model and communities. *Soft Computing*, 20(2) :727–747, 2016.

[157] Weiwu Ren, Jianfei Zhang, Xiaoqiang Di, Yinan Lu, Bochen Zhang, and Jianping Zhao. Anomaly detection algorithm based on cfsfdp. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 24(4) :453–460, 2020.

[158] Wentao Wu and Hong Qin. Reducing noise for pic simulations using kernel density estimation algorithm. *Physics of Plasmas*, 25(10), 2018.

[159] Amit Deshpande, Praneeth Kacham, and Rameshwar Pratap. Robust k-means++. In *Conference on uncertainty in artificial intelligence*, pages 799–808. PMLR, 2020.

[160] SK Sathya Lakshmi Preetha, R Dhanalakshmi, and Rajagopal Kumar. An energy efficient framework for densely distributed wsns iot devices based on tree based robust cluster head. *Wireless Personal Communications*, 103(4) :3163–3180, 2018.

[161] Tianle Zhang, Ali Hassan Sodhro, Zongwei Luo, Noman Zahid, Muhammad Wasim Nawaz, Sandeep Pirbhulal, and Muhammad Muzammal. A joint deep learning and internet of medical things driven framework for elderly patients. *IEEE Access*, 8 :75822–75832, 2020.

[162] Aafreen Qureshi, Shivangi Batra, Prashant Vats, Sandeep Singh, Manu Phogat, and Anupam Kumar Sharma. A review of machine learning (ml) in the internet of medical things (iomt) in the construction of a smart healthcare structure. *Journal of Algebraic Statistics*, 13(2) :225–231, 2022.

[163] Ayoub Si-Ahmed, Mohammed Ali Al-Garadi, and Narhimene Boustia. Survey of machine learning based intrusion detection methods for internet of medical things. *Applied Soft Computing*, 140 :110227, 2023.

[164] Noviyanti TM Sagala and Alexander Agung Santoso Gunawan. Discovering the optimal number of crime cluster using elbow, silhouette, gap statistics, and nbclust methods. *ComTech : Computer, Mathematics and Engineering Applications*, 13(1) :1–10, 2022.

[165] Luyao Li, Yang Qiu, and Jing Xu. A k-means clustered routing algorithm with location and energy awareness for underwater wireless sensor networks. In *Photonics*, volume 9, page 282. MDPI, 2022.

[166] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8 :80716–80727, 2020.

[167] Bryan Raj, Ismail Ahmedy, Mohd Yamani Idna Idris, and Rafidah Md. Noor. A survey on cluster head selection and cluster formation methods in wireless sensor networks. *Wireless Communications and Mobile Computing*, 2022 :1–53, 2022.

[168] Nima Sammaknejad, Yujia Zhao, and Biao Huang. A review of the expectation maximization algorithm in data-driven process identification. *Journal of process control*, 73 :123–136, 2019.

[169] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm : past, present, and future. *Multimedia Tools and Applications*, 80(5) :8091–8126, 2021.

[170] Ranida Hamidouche, Zibouda Aliouat, and Abdelhak Mourad Gueroui. Genetic algorithm for improving the lifetime and qos of wireless sensor networks. *Wireless Personal Communications*, 101(4) :2313–2348, 2018.

[171] Anupriya Shukla, Hari Mohan Pandey, and Deepti Mehrotra. Comparative review of selection techniques in genetic algorithm. In *2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*, pages 515–519. IEEE, 2015.

[172] Padmavathi Kora and Priyanka Yadlapalli. Crossover operators in genetic algorithms : A review. *International Journal of Computer Applications*, 162(10), 2017.

[173] Muhammad Ali Jamshed, Kamran Ali, Qammer H Abbasi, Muhammad Ali Imran, and Masood Ur-Rehman. Challenges, applications and future of wireless sensors in internet of things : A review. *IEEE Sensors Journal*, 2022.

[174] Nihar Ranjan Roy and Pravin Chandra. Energy dissipation model for wireless sensor networks : a survey. *International Journal of Information Technology*, 12(4) :1343–1353, 2020.

[175] Abhijeet Das and Abhishek Swaroop. Energy efficient routing protocol for linear wireless sensor network. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 703–707. IEEE, 2017.

[176] Nazri Mohd Nawi, Muhammad Aamir, Muhammad Faheem Mushtaq, et al. The comparative performance analysis of clustering algorithms. In *International Conference on Soft Computing and Data Mining*, pages 341–352. Springer, 2022.

[177] Shengbo Eben Li. *Deep Reinforcement Learning*, pages 365–402. Springer Nature Singapore, Singapore, 2023.

[178] Maoli Wang, Bowen Zhang, Xiaodong Zang, Kang Wang, and Xu Ma. Malicious traffic classification via edge intelligence in iiot. *Mathematics*, 11(18) :3951, 2023.

[179] Bushra Mohammed, Mosab Hamdan, Joseph Stephen Bassi, Haitham A Jamil, Suleman Khan, Abdallah Elhigazi, Danda B Rawat, Ismahani Binti Ismail, and Muhammad Nadzir Marsono. Edge computing intelligence using robust feature selec-

tion for network traffic classification in internet-of-things. *IEEE Access*, 8 :224059–224070, 2020.

[180] Tosiron Adegbija, Roman Lysecky, and Vinu Vijay Kumar. Right-provisioned iot edge computing : An overview. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, pages 531–536, 2019.

[181] Mahmoud Abbasi, Amin Shahraki, and Amir Taherkordi. Deep learning for network traffic monitoring and analysis (ntma) : A survey. *Computer Communications*, 170 :19–41, 2021.

[182] Stewart Kirubakaran Selvakumar, V Ashok, Jason J Selvadurai, V Ebenezer, Antony J Nivin, and Dennies A Aron. Divination of stock market exploration using long short-term memory (lstm). *Multidisciplinary Science Journal*, 6(1) :2024001–2024001, 2024.

[183] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53 :5929–5955, 2020.

[184] Derya Soydaner. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13) :2052013, 2020.

[185] Paul Fergus and Carl Chalmers. *Performance Evaluation Metrics*, pages 115–138. Springer International Publishing, Cham, 2022.

[186] Dataset of legitimate iot data variot, 2021. Récupérée le 31 décembre 2023 sur le site `https://www.data.gouv.fr/en/datasets/dataset-of-legitimate-iot-data/`.

[187] Yizheng Wang, Yang Yu, Tengfei Zhang, Keyu Song, Yirui Wang, and Shangce Gao. Improved dendritic learning : Activation function analysis. *Information Sciences*, page 121034, 2024.

[188] Timothy O Hodson. Root-mean-square error (rmse) or mean absolute error (mae) : When to use them or not. *Geoscientific Model Development*, 15(14) :5481–5487, 2022.

[189] Ali R Abdellah, Volkov Artem, Ammar Muthanna, Denis Gallyamov, and Andrey Koucheryavy. Deep learning for iot traffic prediction based on edge computing. In *Distributed Computer and Communication Networks : Control, Computation,*

Communications : 23rd International Conference, DCCN 2020, Moscow, Russia, September 14-18, 2020, Revised Selected Papers 23, pages 18–29. Springer, 2020.