

Mémoire de fin d'étude

Pour l'obtention du diplôme d'ingénieur d'Etat en Informatique
Option : Systèmes Informatiques et Logiciels (SL)

VERS UN APPRENTISSAGE FÉDÉRÉ ENTIÈREMENT DISTRIBUÉ POUR L'INTERNET DES OBJETS

Réalisé par :
Hiba Akli
Rima Zourane

Encadré par :
Dr. Mohammed Riyadh Abdmeziem, ESI

Organisme d'accueil :
Laboratoire de Méthodes de Conception des Systèmes LMCS

Promotion : 2022/2023

Remerciements

Nous tenons tout d'abord à exprimer notre gratitude envers Dieu le Tout-Puissant et Miséricordieux, qui nous a donné la force et la patience nécessaires pour accomplir ce modeste travail.

Nous souhaitons en particulier remercier Mr Mohammed Riyadh ABDMEZIEM, notre promoteur et encadrant au sein du laboratoire LMCS. Grâce à lui, nous avons pu bénéficier d'une opportunité unique et nous sommes reconnaissantes pour son soutien, sa gentillesse et ses conseils tout au long de ce travail. Nous tenons également à remercier l'ensemble du personnel de l'École Nationale Supérieure d'Informatique pour leur travail admirable et leur contribution quotidienne à l'excellence de notre école.

Nous voulons également exprimer notre reconnaissance envers les membres du jury, qui ont pris le temps d'évaluer ce modeste travail. Nous sommes conscientes que leur expertise et leur attention nous ont permis de progresser et de nous améliorer.

Nous souhaitons également remercier tous nos amis et toutes les personnes qui nous ont soutenus tout au long de ce travail. Leur encouragement et leur soutien ont été d'une grande aide pour nous.

Enfin, nous tenons à remercier nos familles surtout nos parents, qui ont été une source constante de soutien et d'encouragement pour nous tout au long de nos études. C'est grâce à eux que nous avons pu arriver à ce stade, et nous leur en sommes très reconnaissantes.

Encore une fois, nous sommes conscientes de la chance que nous avons eu de pouvoir mener à bien ce projet, et nous remercions tous ceux qui ont contribué à sa réussite.

Résumé

L'apprentissage fédéré a émergé comme une réponse aux défis posés par la gestion des flux massifs de données générées par l'IoT. Cependant, malgré ses avantages en matière de confidentialité et de préservation de la propriété des données, l'apprentissage fédéré présente des failles en termes de sécurité et disponibilité. Notre solution repose sur l'intégration de la blockchain, une technologie de registre numérique décentralisé qui enregistre et sécurise les transactions de manière transparente et immuable, offrant ainsi une base solide pour construire un système robuste, transparent et de confiance. De plus, afin de garantir la qualité et la fiabilité des résultats, nous mettons l'accent sur l'importance de la sélection rigoureuse des participants.

Cet ouvrage se structure en cinq chapitres, explorant les fondements de l'IoT, de l'apprentissage automatique, de l'apprentissage fédéré et de la blockchain. Il examine également un large éventail d'approches de sélection des nœuds, les classant en deux catégories : les méthodes basées sur des critères prédéfinis pour évaluer les participants tels que leurs ressources, la qualité de leurs modèles ou bien leurs comportements, et les approches reposant sur l'intelligence artificielle, en particulier l'apprentissage par renforcement. Notre solution propose un système d'apprentissage fédéré distribué intégrant un mécanisme de sélection des nœuds hybride en deux étapes, combinant efficacement les avantages de ces deux catégories d'approches.

En outre, notre contribution intègre un mécanisme d'agrégation à plusieurs niveaux pour traiter l'asynchronisme inhérent aux environnements IoT diversifiés. Les évaluations empiriques démontrent que cette approche renforce à la fois les performances et la sécurité, établissant ainsi un cadre solide pour la sélection des nœuds dans le contexte de l'apprentissage fédéré appliquée à l'IoT basé sur la blockchain.

Mots-clés : Apprentissage Fédéré, Internet des Objets, Blockchain, Sélection des noeuds, Agrégation par niveaux, Apprentissage par Renforcement Profond.

Abstract

Federated learning has emerged as a response to the challenges posed by the massive data flows generated by the IoT. However, despite its advantages in terms of confidentiality and privacy preserving, federated learning has its shortcomings in terms of security and availability. Our solution is based on the integration of the Blockchain, a decentralized digital ledger technology that records and secures transactions transparently and immutably, providing a solid foundation for building a robust, transparent and trusted system. In addition, to guarantee the quality and reliability of results, we emphasize the importance of rigorous participant selection.

This work is structured into five chapters, exploring the foundations of IoT, machine learning, federated learning and blockchain. It also examines a wide range of node selection approaches, classifying them into two categories : methods based on predefined criteria for evaluating participants, such as their resources, the quality of their models or their behaviors, and approaches based on artificial intelligence, in particular reinforcement learning. Our solution proposes a distributed federated learning system incorporating a two-stage hybrid node selection mechanism, effectively combining the advantages of both categories.

In addition, our contribution incorporates a multi-level aggregation mechanism to deal with the asynchronism inherent in diverse IoT environments. Empirical evaluations demonstrate that this approach boosts both performance and security, thus establishing a solid framework for node selection in the context of federated learning applied to blockchain-based IoT.

Keywords : Federated Learning, Internet of Things, Blockchain, Node selection, multi-level aggregation, Deep Reinforcement Learning.

ملخص

التعلم الموحد ظهر كاستجابة لتحديات إدارة التدفق الضخم للبيانات التي تُنتجها الإنترنت من الأشياء. ومع ذلك، على الرغم من مزاياه فيما يتعلق بالخصوصية والمحافظة على ملكية البيانات، فإن التعلم الموحد يظهر نقاط ضعف فيما يتعلق بالأمان والتوفير. الحل الذي نقدمه يعتمد على دمج سلسلة الكتل، وهي تكنولوجيا سجل رقمي موزع يُسجل ويؤمن العمليات بطريقة شفافة وثابتة، مما يوفر أساساً قوياً لبناء نظام قوي وشفاف وجدير بالثقة. بالإضافة إلى ذلك، نضع التركيز على أهمية اختيار الأجهزة المشاركة بدقة لضمان جودة وموثوقية النتائج.

تتألف هذه الدراسة من خمسة فصول، حيث تستكشف أسس الإنترنت الأشياء والتعلم الآلي والتعلم الموحد وسلسلة الكتل. كما تقوم بفحص مجموعة واسعة من النهج لاختيار الأجهزة المشاركة، حيث تُصنّف إلى فئتين : الأساليب القائمة على معايير محددة مُسبقاً لتقييم المشاركين مثل مواردهم وجودة نماذجهم أو سلوكهم، والنهج القائمة على الذكاء الإصطناعي، وبشكل خاص تعلم المعزز العميق. تشتمل مساحتنا على نظاماً للتعلم الموحد موزعاً يدّفع آلية اختيار الأجهزة المشاركة في مرحلتين، مما يجمع بفعالية بين مزايا هاتين الفئتين من النهج.

بالإضافة إلى ذلك، تضمن مساحتنا آلية تجميع متعدد المستويات لمعالجة التفاوت الزمني في بيئات الإنترنت من الأشياء المتعددة. تظهر التقييمات التجريبية أن هذا النهج يعزز الأداء والأمان على حد سواء، وبالتالي يوفر إطاراً قوياً لاختيار الأجهزة المشاركة في سياق التعلم الموحد المطبق على الإنترنت الأشياء مبني على سلسلة الكتل.

الكلمات المفتاحية : التعلم الموحد، إنترنت الأشياء، سلسلة الكتل، اختيار الأجهزة المشاركة، التجميع متعدد المستويات، التعلم المعزز العميق

Sommaire

Remerciement	I
Résumé	II
Abstract	III
Sommaire	VII
Liste des figures	VIII
Liste des tableaux	X
Liste des sigles et abréviations	XII
Introduction générale	1
I Synthèse bibliographique	3
1 Internet des Objets et Apprentissage Automatique	4
1.1 Introduction	4
1.2 Internet des Objets	5
1.2.1 Définition	5
1.2.2 Caractéristiques	5
1.2.3 Architectures et protocoles	6
1.2.4 Défis	8
1.3 Apprentissage Automatique	10
1.3.1 Définition	10
1.3.2 Types	11
1.3.3 Fonctionnement	14
1.3.4 Défis	15
1.4 Conclusion	16
2 Apprentissage Fédéré et Blockchain	18
2.1 Introduction	18
2.2 Apprentissage Fédéré	18

SOMMAIRE

2.2.1	Définition	18
2.2.2	Fonctionnement de l'apprentissage fédérée	19
2.2.3	Types d'apprentissage fédéré	20
2.2.4	Défis	24
2.2.5	Application de l'apprentissage fédéré pour IoT	25
2.3	Blockchain	27
2.3.1	Définition	27
2.3.2	Types de Blockchain	28
2.3.3	Caractéristiques	29
2.3.4	Algorithmes de consensus	30
2.4	Conclusion	33
3	État de l'art	34
3.1	Introduction	34
3.2	Méthodologie de recherche	35
3.3	Travaux relatifs à la sélection des nœuds	36
3.3.1	Approches Algorithmiques	37
3.3.2	Approches intelligentes	43
3.4	Comparaison et évaluation des approches	47
3.4.1	Métriques d'évaluation utilisées	47
3.4.2	Comparaison des approches	49
3.4.3	Analyse et discussion	52
3.5	Conclusion	53
II	Contribution	54
4	Conception de la solution	55
4.1	Introduction	55
4.2	Architecture globale du système	55
4.2.1	Processus de travail	57
4.2.2	Modèles de communication	60
4.2.3	Modèles adversaires	60
4.3	Métriques de sélection des nœuds	62
4.3.1	Contribution	62
4.3.2	Score d'honnêteté	63
4.3.3	Coût global	64
4.4	Mécanisme hybride distribué pour la sélection des nœuds	64
4.4.1	Méthode basée sur le score	65
4.4.2	Méthode DRL	66
4.4.3	Méthode Hybride	69
4.5	Agrégation globale à plusieurs niveaux du FL	70
4.5.1	Processus d'agrégation	70
4.5.2	Évaluation des modèles	72
4.6	Conclusion	72

SOMMAIRE

5 Réalisation et évaluation	75
5.1 Introduction	75
5.2 Présentation des outils utilisés	75
5.2.1 Langages de programmation	75
5.2.2 Bibliothèques et framework	76
5.3 Détails de l'implémentation	77
5.3.1 Environnement de simulation	77
5.3.2 Architecture des réseaux de neurones et dataset utilisés	78
5.3.3 Paramétrage	79
5.4 Évaluation	81
5.4.1 Sélection basée sur le score	81
5.4.2 Sélection basée sur le DRL	87
5.4.3 Sélection hybride	90
5.4.4 Comparaison et Synthèse	94
5.5 Conclusion	98
Conclusion Générale	99
Bibliographie	101
Webographie	108
Annexe	110

Liste des figures

1.1	Pile IoT générique comparée au modèle OSI et le modèle TCP/IP (TOURNIER et al., 2021)	7
1.2	Pile des protocoles IoT (JAGANNATH et al., 2019)	8
1.3	Apprentissage automatique et ses types (GÉRON, 2022 ; MAHESH, 2019 ; RAVISHANKAR & VIJAYAKUMAR, 2017)	11
2.1	Les différentes catégorisation d'apprentissage fédéré (Q. LI et al., 2021 ; C. ZHANG et al., 2021)	20
2.2	Les types d'apprentissage fédéré selon le partitionnement des données (NGUYEN et al., 2021)	21
2.3	Les types d'apprentissage fédéré selon l'architecture de communication (NGUYEN et al., 2021)	23
2.4	La structure de la Blockchain (LIANG, 2020)	28
2.5	Proof of Work (KOLB et al., 2020)	30
3.1	Taxonomie des travaux relatifs à la sélection des nœuds	36
3.2	Mesure de la contribution (J. ZHANG et al., 2021)	38
3.3	Comparaison des valeurs de réputation et de récompenses	40
3.4	Protocol FedCS (NISHIO & YONETANI, 2018)	41
3.5	Exemple d'exécution de l'algorithme heuristique du problème de la secrétaire (MOHAMMED et al., 2020)	42
3.6	Architecture d'apprentissage fédéré distribué avec sélection DRL des nœuds	44
3.7	Le mécanisme hybride de blockchain PermiDAG (LU et al., 2020)	45
4.1	Architecture globale du système	58
4.2	Mécanisme hybride distribué pour la sélection des nœuds	65
4.3	Modèle d'apprentissage par renforcement profond du type Actor-Critic . .	68
4.4	Fonctionnement du mécanisme hybride distribué pour la sélection des nœuds	70
4.5	Exemple d'agrégation	71
4.6	Diagramme de séquence du processus d'évaluation et d'agrégation	73
5.1	Performance de la machine utilisée en terme de CPU	77
5.2	Performance de la machine utilisée en terme de RAM	78
5.3	Convergence du modèle global avec sélection basée sur le score avec 50 nœuds et variation du paramètre α	82

LISTE DES FIGURES

5.4 Convergence du modèle global avec sélection basée sur le score avec 150 nœuds et variation du paramètre α	82
5.5 Convergence du modèle global avec sélection basée sur le score avec 300 nœuds et variation du paramètre α	83
5.6 Les valeurs du score d'honnêteté des nœuds avec sélection basée sur le score.	84
5.7 Convergence du modèle global avec sélection basée sur le score avec différents pourcentages de nœuds malveillants	85
5.8 Convergence du modèle global avec sélection basée sur le score avec différents pourcentages d'abandon	86
5.9 Convergence du modèle global avec sélection DRL avec différents nombre de nœuds.	87
5.10 Les valeurs du score d'honnêteté des nœuds avec sélection DRL.	88
5.11 Convergence du modèle global avec sélection DRL avec différents pourcentages de nœuds malveillants.	89
5.12 Convergence du modèle global avec sélection DRL avec différents pourcentages d'abandon.	90
5.13 Convergence du modèle global avec sélection hybride avec différents nombres de nœuds.	91
5.14 Les valeurs du score d'honnêteté des nœuds avec sélection hybride.	92
5.15 Convergence du modèle global avec sélection hybride avec différents pourcentages de nœuds malveillants.	93
5.16 Convergence du modèle global avec sélection hybride avec différents pourcentages d'abandon.	93
5.17 Convergence du modèle global avec les trois approches dans un réseau de 300 nœuds.	95
5.18 Convergence du modèle global avec les trois approches dans un réseau avec un taux de nœuds malhonnêtes de 60%.	95
5.19 Convergence du modèle global avec les trois approches dans un réseau avec un taux d'abandon de 60%.	96
5.20 Temps d'exécution des trois approches de sélection avec différents taux d'abandon et nœuds malveillants.	96
21 Exemple de machine à vecteur support	111
22 L'effet du taux d'apprentissage sur la descente du gradient	113

Liste des tableaux

3.1 critères de sélection des noeuds	50
3.2 Tableau comparatif des approches étudiées	51
4.1 Sommaire des notations principales	56
5.1 Paramètres de simulation	80
5.2 Autres paramètres de Simulation	80
5.3 Paramètres du scénario 1 pour la sélection basée sur le score	81
5.4 La précision finale du modèle global du scénario 1 avec sélection basée sur le score	83
5.5 Paramètres du scénario 2 pour la sélection basée sur le score	84
5.6 La précision finale du modèle global du scénario 2 avec sélection basée sur le score	85
5.7 Paramètres du scénario 3 pour la sélection basée sur le score	85
5.8 La précision finale du modèle global du scénario 3 avec sélection basée sur le score	86
5.9 Paramètres du scénario 1 pour la sélection DRL	87
5.10 La précision finale du modèle global du scénario 1 avec sélection par DRL . .	87
5.11 Paramètres du scénario 2 pour la sélection DRL	88
5.12 La précision finale du modèle global du scénario 2 avec sélection par DRL . .	89
5.13 Paramètres du scénario 3 pour la sélection DRL	89
5.14 La précision finale du modèle global du scénario 3 avec sélection par DRL . .	90
5.15 Paramètres du scénario 1 pour la sélection hybride	91
5.16 La précision finale du modèle global du scénario 1 avec sélection hybride . .	91
5.17 Paramètres du scénario 2 pour la sélection hybride	92
5.18 La précision finale du modèle global du scénario 2 avec sélection hybride . .	92
5.19 Paramètres du scénario 3 pour la sélection hybride.	93
5.20 La précision finale du modèle global du scénario 3 avec sélection hybride . .	94

Liste des sigles et abréviations

- BA** Byzantine Agreement
BFT Byzantine Fault Tolerance
CCPA California Consumer Privacy Act
CNN Convolutional Neural Network
DNN Deep Neural Network
DP Differential Privacy
DRL Deep Reinforcement Learning
FNN Feed Forward Neural Network
FL Federated Learning
GDPR General Data Protection Regulation
HFL Horizontal Federated Learning
IA Intelligence Artificielle
IoT Internet of Things
IETF Internet Engineering Task Force
LSTM Long Short Term Memory
LoRaWAN Low Range Wide Area Network
MBS Macro Base Station
MDP Markov Decision Process
MEC Mobile Edge Computing
ML Machine Learning
Non-IID Non Independent and Identically Distributed
PDPA Personal Data Protection Act
PBFT Practical Byzantine Fault Tolerance
PoA Proof of Authority
PoET Proof of Elapsed Time
PoS Proof of Stake
PoW Proof of Work
RFID Radio-frequency identification

LISTE DES TABLEAUX

RL	Reinforcement Learning
SVM	Support Vector Machines
SMC	Secure Multi-party Computation
TCAC	Taux de croissance annuel composé
TEE	Trusted Execution Environments
VFL	Vertical Federated Learning
6LoWPAN	IPv6 Low Wide Personal Area Network
P2P	Peer to Peer

Introduction générale

Au cours de la dernière décennie, l'Internet des Objets (IoT) a connu une croissance exponentielle et s'est étendue à un large éventail d'applications, allant des appareils domestiques aux usines intelligentes et aux villes intelligentes. Cette prolifération de l'IoT a été alimentée par la disponibilité de dispositifs connectés à Internet et par la demande croissante de données en temps réel. L'intégration de l'apprentissage automatique dans les environnements IoT était une nécessité pour améliorer leur efficacité et leur performance. Cependant, cette intégration a également introduit de nouveaux problèmes, notamment la mise en péril de données confidentielles, qui ont conduit à l'émergence de l'apprentissage fédéré.

L'apprentissage fédéré est une méthode d'intelligence artificielle distribuée. Il permet à un ensemble de machines distribuées d'entraîner un modèle sans le partage de données. En effet, malgré les avantages de l'apprentissage fédéré, il présente également plusieurs vulnérabilités. L'une des vulnérabilités majeures est le serveur central d'agrégation. Étant donné que tous les modèles sont agrégés sur un seul serveur central, il représente un point de défaillance unique. Si ce serveur est corrompu, cela peut compromettre la confidentialité et la sécurité des données des usagers.

De plus, les participants de l'apprentissage fédéré peuvent également être une vulnérabilité. Les appareils IoT ont souvent des ressources limitées, ce qui peut dégrader la performance globale du système. Par ailleurs, ils peuvent être la cible des attaques telles que l'injection de données malveillantes ou le piratage. En utilisant la blockchain, une technologie de stockage et de transmission d'informations sans autorité centrale, pour distribuer le processus d'apprentissage fédéré, la transparence, la disponibilité et la sécurité sont garanties. Cependant, cela ne résout pas le problème présenté par la vulnérabilité des participants. Il est alors important de choisir soigneusement les participants pour garantir la qualité et la sécurité des résultats, en particulier lorsqu'il s'agit d'appareils IoT avec des ressources limitées.

La sélection rigoureuse des nœuds pour l'apprentissage fédéré est cruciale pour garantir la qualité, la sécurité et la fiabilité des résultats dans les environnements IoT. Les approches de sélection des nœuds doivent prendre en compte des facteurs tels que la capacité de calcul et de communication, la disponibilité, la fiabilité et la sécurité des composantes du système pour garantir une performance optimale et une confidentialité maximale des données. Avec la prolifération de l'IoT et de la demande croissante des

LISTE DES TABLEAUX

données en temps réel, le développement de solutions d'apprentissage fédéré pour l'IoT et la sélection des noeuds deviennent de plus en plus importants pour répondre aux défis de confidentialité et de fiabilité posés par cette technologie en constante évolution.

De nombreuses recherches ont été menées pour développer des approches efficaces pour la sélection des noeuds dans les systèmes d'apprentissage fédéré pour l'IoT. Différentes méthodes ont été proposées, chacune ayant ses avantages et ses inconvénients.

Le premier objectif de notre travail est de mener une exploration approfondie des approches récentes pour la sélection des nœuds dans les environnements d'apprentissage fédéré dédiés à l'Internet des Objets. Notre démarche vise également à apporter une contribution novatrice à ce domaine en proposant une solution qui adresse les problématiques identifiées.

Cette contribution comporte une nouvelle approche de sélection des nœuds, qui prend en compte le comportement, la contribution et les ressources des participants. Par ailleurs, nous proposons également une distribution entière du processus de l'apprentissage fédéré, avec une agrégation à plusieurs niveaux et en intégrant la blockchain. De plus, afin de renforcer la sécurité du système, nous mettons en place plusieurs étapes de vérification et de validation des modèles.

Ce mémoire comporte cinq chapitres. Le premier chapitre traitera de l'Internet des Objets et de l'apprentissage automatique, en abordant les caractéristiques, les protocoles et les défis. Le deuxième chapitre présentera l'Apprentissage Fédéré et la Blockchain, ainsi que leur intégration avec l'IoT. Le troisième chapitre passera en revue plusieurs travaux récents sur la sélection des nœuds, présentant un état de l'art, ainsi qu'une évaluation et une comparaison de ces approches. Le quatrième chapitre présente notre solution, en détaillant les différentes contributions proposées. Enfin, le dernier chapitre explique les choix d'implémentation et les détails de réalisation pour ensuite évaluer les résultats obtenus à travers une variété de scénarios.

Première partie

Synthèse bibliographique

Chapitre 1

Internet des Objets et Apprentissage Automatique

1.1 Introduction

L'Internet des objets (IoT) représente l'intégration des objets physiques à l'Internet (FAROOQ et al., [s. d.](#)). L'Intelligence Artificielle (IA) quant à elle, s'intéresse au développement de systèmes intelligents capables d'agir de manière appropriée dans des environnements dynamiques (FATIMA et al., [2020](#)). Une intersection évidente existe entre ces deux domaines. l'IoT fournit une infrastructure matérielle et logicielle, offrant un accès à toutes les données générées par ses dispositifs. Elle permet aussi d'entreprendre des actions pour faire évoluer l'environnement de déploiement . Ces données ne sont utiles que si elles permettent de conduire à des actions, cependant elles sont de nature hétérogène et très complexe, due à la diversité des objets IoT, et donc les approches traditionnelles de traitement de données ne sont pas suffisamment sophistiquées pour pouvoir exploiter les données collectées. Dans ce contexte, l'IA est nécessaire pour traiter les volumes exorbitants de données générées, en extraire un sens, raisonner sur ces données et décider des actions appropriées. Pour que ces données et ces actions soient exploitables, les domaines de l'IoT et de l'IA doivent travailler en synergie dans le but de passer de systèmes d'objets connectés à des systèmes d'intelligences connectées, et cela dans différents domaines d'applications, par exemple un système de gouvernement intelligent (KANKANHALLI et al., [2019](#)), les soins médicaux intelligents (ALSHEHRI & MUHAMMAD, [2020](#)) ou bien dans le domaine de l'agriculture (MISRA et al., [2020](#)).

Dans ce chapitre, nous allons présenter les différents concepts liés à l'IoT et l'apprentissage automatique, qui est le sous domaine de l'intelligence artificielle le plus pertinent pour notre étude. Concernant l'Internet des Objets, nous allons voir sa définition, ses caractéristiques, les architectures, les différents protocoles de communication et les challenges. Viendra par la suite la deuxième section qui traitera l'Apprentissage Automatique, ses types, son fonctionnement et ses challenges.

1.2 Internet des Objets

1.2.1 Définition

L'Internet des objets (IoT) est en train de changer notre façon de vivre, de travailler, de voyager et de faire du business. C'est même la base d'une transformation industrielle, connue sous le nom d'Industrie 4.0 (HERMANN et al., 2015), et la clé de la transformation numérique des organisations (LEE, 2017), des villes et de la société en général (GAUR et al., 2015).

Le concept d'internet des objets a été introduit pour la première fois par Kevin Ashton en 1999, il l'a définit en tant qu'objets connectés interopérables à identification unique par radiofréquence. Depuis, cette définition a évoluée pour aboutir à ce que l'IoT représente de nos jours “une infrastructure de réseau mondiale dynamique dotée de capacités d'auto-configuration fondées sur des normes et des protocoles de communication interopérables, les objets physiques et virtuels d'un IoT possèdent des identités et des attributs et sont capables d'utiliser des interfaces intelligentes et d'être interconnectés en tant que réseau d'information” (S. LI et al., 2015). Pour faire simple, l'internet des objets est un réseau mondial d'objets uniquement identifiables, interconnectés et qui interagissent et coopèrent entre eux de façon autonome, sans l'interférence des utilisateurs, afin de réaliser un objectif commun.

1.2.2 Caractéristiques

L'IoT est un domaine à contraintes, où son déploiement nécessite une architecture et une conception bien définies qui répondent à ses caractéristiques. Ces dernières peuvent être résumées comme suit (ABDMEZIEM et al., 2016 ; PATEL et al., 2016) :

- **Interconnectivité** : tout objet faisant partie de l'IoT est interconnecté avec une infrastructure mondiale d'information et de communication.
- **Connectivité** : la connectivité permet l'accessibilité et la compatibilité des réseaux. L'accessibilité est le fait d'accéder à un réseau, tandis que la compatibilité fournit la capacité commune de consommer et de produire des données.
- **Hétérogénéité** : les dispositifs de l'IoT sont hétérogènes car ils reposent sur des plateformes matérielles et des réseaux différents. Ils peuvent interagir avec d'autres appareils ou plateformes de services par l'intermédiaire de différents réseaux.
- **Distributivité** : l'IoT évolue dans un environnement hautement distribué. En effet, les données peuvent être collectées à partir de différentes sources et traitées par plusieurs entités de manière distribuée.
- **Changement dynamique** : l'état des appareils change de manière dynamique, par exemple en dormant et en se réveillant, en étant connecté et/ou déconnecté, ainsi que le contexte des appareils, notamment leur emplacement et leur vitesse. En outre, le nombre d'appareils dans un réseau peut changer de manière dynamique.

1.2.3 Architectures et protocoles

Plus de 29 milliards de dispositifs IoT sont estimés à être connectés d'ici 2030¹, cela représente un chiffre énorme, qui nécessite une architecture ouverte capable de résoudre divers problèmes de sécurité et de qualité de service (QoS), tout en prenant en charge les applications réseau existantes à l'aide de protocoles ouverts.

Architectures

Aujourd'hui, des dizaines d'architectures et de protocoles IoT sont proposés par la communauté des chercheurs, et de nombreux défis ont été identifiés, notamment l'interopérabilité, la sécurité et la confidentialité, la fiabilité, les contraintes des ressources, l'évolutivité et l'absence de normes communes, tous ces challenges font que toutes les architectures proposées et utilisées ne sont pas suffisamment matures pour être standardisées (AL-QASEEMI et al., 2016). Pour fournir une architecture IoT appropriée, l'importance et la priorité des exigences peuvent varier selon les scénarios et les cas d'application de l'IoT, et donc une analyse des exigences doit être considérée comme un élément essentiel (SAMIZADEH NIKOUI et al., 2021).

La plupart des architectures proposées sont des architectures multicouches : 3 couches (application, réseau, perception) (ABDMEZIEM et al., 2016 ; AL-QASEEMI et al., 2016), 4 couches parmi lesquels nous citons l'architecture SOA (Service Oriented Architecture) (S. LI et al., 2015 ; PATEL et al., 2016 ; SUO et al., 2012), 5 couches (AL-QASEEMI et al., 2016) et 6 couches (FAROOQ et al., s. d.). Toutes ces architectures sont des adaptations des standards de communication sur internet, les modèles OSI et TCP/IP, afin de répondre aux besoins et contraintes des systèmes IoT. Les spécificités de l'IoT, telles que la rareté des ressources, l'instabilité des liens sans fil et l'hétérogénéité du trafic et des dispositifs, vont sérieusement entraver le déploiement des protocoles IP dans les environnements IoT (ABDMEZIEM et al., 2016). La figure 1.1 représente une comparaison d'un modèle générique pour l'IoT proposé par (TOURNIER et al., 2021) avec les deux suites de protocoles OSI et TCP/IP.

La pile générique IoT présentée dans la figure 1.1 est composée de cinq couches, où chaque couche représente des fonctionnalités spécifiques : la couche Physique et Lien de Données spécifient les fonctionnalités de radio-fréquence, la couche Réseau définit le routage et les protocoles de sécurité, et les deux dernières couches, Transport et Application, spécifient les commandes disponibles dans les protocoles (TOURNIER et al., 2021).

Protocoles

D'une part, les systèmes IoT doivent être capables de faire face à des connexions potentiellement peu fiables et à faible bande passante pour leurs réseaux d'accès, et d'autre part, ces objets, entièrement hétérogènes, doivent communiquer d'une manière transparente et commune. A fin de réussir à surmonter ces challenges, plusieurs protocoles de communication ont été proposés (JAGANNATH et al., 2019). Parmi ceux-ci, citons ZigBee, 6LoWPAN (IPv6 sur les réseaux personnels sans fils à faible puissance), le protocole

1. <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

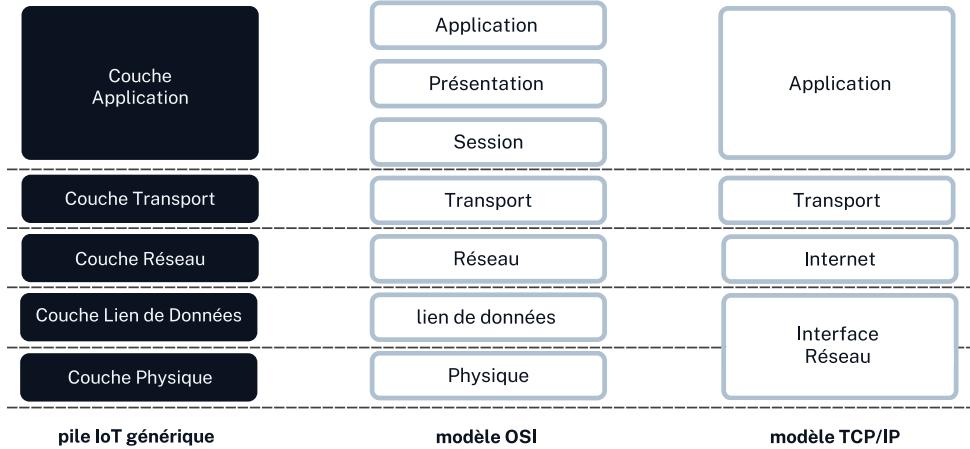


FIGURE 1.1 – Pile IoT générique comparée au modèle OSI et le modèle TCP/IP (TOURNIER et al., 2021).

de routage RPL, Bluetooth basse énergie, WirelessHart, ISA100.11a, LoRAWAN (protocole de réseau étendu à longue portée), NB-IoT(IoT à bande étroite). La figure 1.2 illustre l'emplacement de ces protocoles selon les différentes couches du modèle générique proposé précédemment. Et dans ce qui suit nous définissons les protocoles les plus importants :

- **IEEE 802.15.4** est une norme élaborée par IETF (Internet Engineering Task Force), elle constitue à la fois la couche physique et le contrôle d'accès au support pour les dispositifs sans fil à contraintes (CALLAWAY, 2003). Plusieurs protocoles des couches supérieures sont basées sur IEEE 802.15.4 tels que Zigbee, 6LoWPAN, WirelessHart, MiWi et ISA100.11a (JAGANNATH et al., 2019).
- **6LoWPAN** (IPv6 over Low power Wireless Personal Area Networks) est le protocole le plus utilisé dans les protocoles de communication des IoT, car il permet de transporter des paquets IPv6 sur les réseaux IEEE 802.15.4, et par conséquent une interconnexion entre 6LoWPAN et les réseaux IPv6 classiques. L'utilisation d'IPv6 permet à 6LoWPAN d'exploiter ses fonctionnalités spécifiques, telles que la découverte de voisins, la résolution d'adresses par multicast scopé par lien local, la détection d'adresses dupliquées et la découverte de routeurs. Toutefois, les paquets IPv6 sont plus volumineux que ceux de la norme IEEE 802.15.4 (TOURNIER et al., 2021).
- **LoRaWAN** (Low Range Wide Area Network) est un protocole de couche MAC, et il est basé sur la communication LoRa (la couche physique), qui a été proposée pour connecter un grand nombre de périphériques dans de grandes zones (jusqu'à 20 km) avec une faible consommation d'énergie. LoRa module les signaux dans une bande ISM sub-GHZ en utilisant une technique de spectre étalé, grâce à laquelle les messages peuvent être envoyés à une plus grande distance au détriment du débit de données (MILES et al., 2020).
- **RFID** (Radio Frequency IDentification) est un moyen de communication à identification unique des objets, par le biais de signaux radios. Cette technologie utilise deux types de dispositifs, les étiquettes et les radars RFID, le radar s'occupe de dé-

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

tecter et interroger les étiquettes, en envoyant des signaux et en recevant la réponse des étiquettes (ABDMEZIEM et al., 2016).

- **MQTT** (Message Queuing Telemetry Transport) est un protocole publication/abonnement basé sur le protocole TCP, développé par IBM ensuite ouvert aux applications de messagerie. Les clients peuvent "publier" des données sur le serveur ou bien "s'abonner" à un sujet où le serveur envoie les nouvelles données aux abonnés (CHEN & KUNZ, 2016).
- **CoAP** (Constrained Application Protocol) Le protocole CoAP est un protocole sans état, développé par l'IETF pour remplacer HTTP dans les dispositifs à ressources limitées. En tant que protocole REST basé sur UDP, il utilise une structure requête/réponse et présente une faible surcharge et un faible degré de QoS optionnelle (CHEN & KUNZ, 2016).
- **DDS** (Data Distributive Service) est un protocole basé sur TCP qui présente des nœuds décentralisés de clients à travers un système et permet à ces nœuds de s'identifier en tant qu'abonnés ou éditeurs par le biais d'une localisation. Il représente un intericiel de mise en réseau pour contourner les inconvénients de l'architecture centralisée de publication et d'abonnement. L'utilisation de ce système évite aux utilisateurs d'avoir à identifier où se trouvent d'autres nœuds potentiels ou les sujets qui les intéressent (CHEN & KUNZ, 2016).

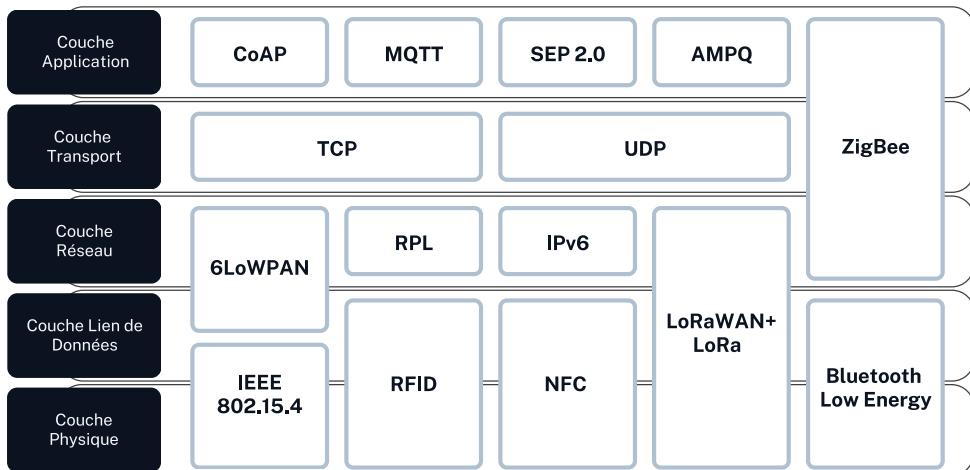


FIGURE 1.2 – Pile des protocoles IoT (JAGANNATH et al., 2019).

1.2.4 Défis

La croissance rapide de l'IoT au cours de ces dernières années a fait naître de nombreux défis qui peuvent être regroupés sous quatre points principaux (PATEL et al., 2016 ; SAMIZADEH NIKOUI et al., 2021) :

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

Limitation des ressources

De nombreux dispositifs IoT ont des ressources limitées en termes d'énergie, de stockage, de communication et de stabilité de la connectivité. De plus, en fonction des applications IoT, comme par exemple un système de surveillance d'un lieu inhabité, les dispositifs peuvent avoir besoin de fonctionner pendant une longue période sans intervention humaine pour les entretenir ou les réparer. Dans de telles circonstances, la défaillance ou l'arrêt d'un dispositif peut réduire la disponibilité du système ou la violation de l'accord de niveau de service. Il est donc nécessaire d'envisager une utilisation efficace des ressources.

Sécurité

Les systèmes IoT ont été utilisés ces dernières années dans de nombreuses applications. Certaines d'entre elles sont des applications critiques telles que l'IoT militaire, les réseaux électriques intelligents, la surveillance de l'environnement et d'autres application qui peuvent contenir des données personnelles ou privées, comme les maison intelligentes, les véhicules intelligents et la santé intelligente. En outre, il existe une grande variété d'attaques qui menacent la sécurité de l'IoT. Par conséquent, l'IoT nécessite un mécanisme de protection tout en satisfaisant les performances du système avec une faible surcharge, ce qui constitue un véritable défi. La sécurité des créateurs et bénéficiaires de l'IoT doit être assurée, et cela inclut la vie privée, l'intégrité, la confidentialité, la disponibilité, l'authenticité, la non-répudiation et la gestion des clés.

Interopérabilité

Dans l'internet traditionnel, l'interopérabilité est le principe de base le plus fondamental, la première exigence de la connectivité internet est que les systèmes connectés soient capables de parler le même langage de protocoles et de codages. Dans un environnement totalement interopérable, tout dispositif devrait être en mesure de se connecter à tout autre dispositif ou système et d'échanger des informations à volonté sans le besoin de connaître les caractéristiques et spécificités des autres dispositifs. Quant à l'IoT, il vise à intégrer le monde physique au monde virtuel en utilisant l'internet comme moyen de communication et d'échange d'informations. Cependant, l'hétérogénéité des dispositifs et des technologies de communication et l'interopérabilité à différents niveaux, allant de la communication et de l'intégration transparente des dispositifs à l'interopérabilité des données générées par les sources de l'IoT, représentent un challenge pour étendre les solutions génériques de l'IoT à grande échelle. Plus particulièrement dans les futurs réseaux, qui continueront à être hétérogènes, avec multifournisseurs, multiservices et largement distribués, le risque de non-interopérabilité augmentera.

Scalabilité

Dans l'IoT, des milliards d'objets sont appelés à faire partie du réseau, et donc les systèmes et les applications qui s'exécutent au-dessus d'eux devront être capable de manipuler cette quantité sans précédent de données générées. Afin d'y parvenir, des approches

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

appropriées devrait être implémentées pour gérer le stockage, les mécanismes de recherche, le traitement et l'analyse de ces données. D'un autre côté, plus le nombre de dispositifs IoT augmente, plus les configurations et la gestion du réseau demandent du temps et des efforts, surtout que ces configurations ne peuvent pas être effectuées manuellement, car cela augmenterait le risque de défaillance. Par conséquent, minimiser ou même éliminer l'intervention humaine et fournir un mécanisme efficace et flexible pour étendre le réseau IoT sont considérés comme des conditions préalables pour atteindre une plus grande scalabilité. En outre, une telle architecture scalable doit fournir un mécanisme d'adressage approprié.

Une étude récente mentionnée dans (L. U. KHAN et al., 2021) a révélé que la taille des données générées par les dispositifs IoT va atteindre 79.4 zettabytes (ZB) en 2025. Une telle croissance dans les réseaux IoT accompagnée par des volumes exorbitants de données offrent beaucoup d'opportunités à l'intelligence artificielle. Cela permettra d'évoluer vers des systèmes IoT plus intelligents et innovants tels les maisons intelligentes, les véhicules autonomes et l'industrie intelligente. En outre, l'IA peut être utilisée pour améliorer différents aspects dans les réseaux IoT (NGUYEN et al., 2021) : la détection d'attaques et d'anomalie, assurer la confidentialité et la sécurité, le partage des données, le déchargement et la mise en cache des données, le crowdsourcing mobile et la localisation . Dans ce qui suit, nous allons définir un sous domaine de l'intelligence artificielle, connu sous l'apprentissage automatique.

1.3 Apprentissage Automatique

1.3.1 Définition

La capacité de créativité du cerveau humain a conduit à l'invention de machines, ce qui a facilité de nombreuses tâches de la vie. Parmi celles-ci figurent des systèmes qui apprennent de leurs expériences passées et forment des idées abstraites à partir de leurs observations.

Selon (SAMUEL, 1988), l'apprentissage automatique est défini comme le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. Il est souvent considéré comme le père de l'apprentissage automatique car son programme de jeu de dames (SAMUEL, 1988) était l'une des premières applications.

L'apprentissage automatique (ML) désigne les méthodes intelligentes utilisées pour optimiser les critères de performance à partir de données d'exemple ou d'expérience(s) passée(s) par le biais de données d'apprentissage. Plus précisément, les algorithmes ML construisent des modèles de comportements en utilisant des techniques mathématiques sur d'énormes ensembles de données (FATIMA et al., 2020). Le ML procure également la capacité d'apprendre sans être explicitement programmé. Ces modèles sont utilisés comme base pour faire des prédictions futures basées sur les données nouvellement entrées. L'apprentissage automatique est interdisciplinaire par nature et hérite de ses racines dans de nombreuses disciplines scientifiques et techniques, notamment l'intelligence artificielle, la théorie de l'optimisation, la théorie de l'information et les sciences cognitives (QIU

1.3.2 Types

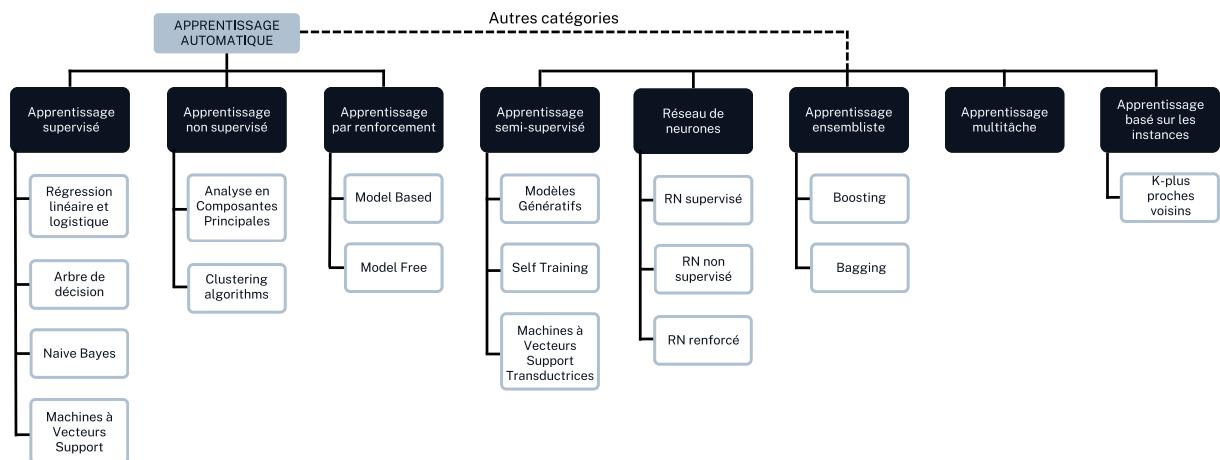


FIGURE 1.3 – Apprentissage automatique et ses types (GÉRON, 2022 ; MAHESH, 2019 ; RAVISHANKAR & VIJAYAKUMAR, 2017).

Les data scientists aiment souligner qu'il n'y a pas de type unique d'algorithme qui soit le meilleur pour résoudre un problème (MAHESH, 2019). Le type d'algorithme utilisé dépend du genre de problème que l'on souhaite résoudre, du nombre de variables et du type de modèle qui lui conviendrait le mieux (FATIMA et al., 2020). La figure 1.3 donne un aperçu de quelques uns des algorithmes les plus utilisés dans l'apprentissage automatique selon (GÉRON, 2022 ; MAHESH, 2019 ; RAVISHANKAR & VIJAYAKUMAR, 2017), et qui sont présentés dans ce qui suit :

Apprentissage supervisé

L'apprentissage supervisé est effectué lorsque des objectifs spécifiques à atteindre sont définis à partir d'un certain ensemble d'entrées. Ce type d'apprentissage peut être divisé sur deux types de problématique : les problèmes de classification et la régression. Les algorithmes de classification permettent d'affecter les données à des classes spécifiques, ils tentent d'identifier automatiquement des règles à partir des ensembles de données d'entraînement et de prédire l'appartenance des données de test aux diverses classes, parmi les algorithmes de classification : classificateurs linéaires, Support Vector Machine (SVM), arbres de décision, forêts aléatoires, K-nearest neighbor (KNN) et Naïve Bayes. En revanche, les algorithmes de régression essaient de comprendre le lien entre les variables dépendantes et indépendantes, ils sont généralement utilisés pour effectuer des projections, par exemple pour prédire le revenu d'un business, parmi ces algorithmes, nous citons : la régression linéaire, la régression logistique et la régression polynomiale².

2. [What is supervised learning?](#) (Consulté le 20 février 2023)

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

Apprentissage non supervisé

Dans l'apprentissage non supervisé, l'environnement ne fournit que des entrées sans cibles souhaitées. L'approche consiste à reconnaître les liens existants entre les données afin d'en déduire des règles et des modèles. Cette technique est appropriée dans une situation où les catégories de données sont inconnues. Dans ce cas, les données d'entraînement ne sont pas étiquetées. Il est considéré comme une approche d'apprentissage basée sur les statistiques et se réfère donc au problème de la recherche de structures cachées dans des données non étiquetées. Parmi les applications d'apprentissage non supervisé, nous citons : Les algorithmes de regroupement (Clustering) tel que le clustering hiérarchique avec K-means, les algorithmes de visualisation qui permettent de représenter des données très complexes en 2D ou 3D et la détection d'anomalie qui est possible grâce à l'extraction des caractéristiques et les différents liens/patterns entre les données.

Apprentissage par renforcement

Le principe de l'apprentissage par renforcement est très différent des autres méthodes ML. Le système d'apprentissage contient un agent qui apprend à prendre les bonnes décisions avec de l'expérience (GÉRON, 2022). Il s'inspire largement des comportements d'apprentissage des humains et des animaux. De tels comportements font une approche attrayante dans les applications hautement dynamiques de la robotique dans lesquelles le système apprend à accomplir certaines tâches sans programmation explicite (DABNEY et al., 2018).

Dans l'apprentissage par renforcement (RL), aucun résultat spécifique n'est défini et l'agent apprend par rétroaction en observant son environnement. Par la suite il effectue une action selon l'état de l'environnement, et en fonction de son action il reçoit un feedback, qui est soit une récompense soit une pénalité. L'objectif de l'agent est d'apprendre une politique (policy), c'est une association (mapping) entre les états et les actions, qui maximise la récompense attendue au fil du temps. A cet effet, il est très important de choisir la fonction de récompense appropriée car le succès et l'échec de l'agent dépendent de la récompense totale accumulée (WIRTH et al., 2017). Cependant, l'un des principaux défis de l'apprentissage par renforcement est de trouver un équilibre entre l'exploration et l'exploitation (KAELBLING et al., 1996) : l'agent doit explorer différentes actions pour apprendre la meilleure politique, mais il doit également exploiter ses connaissances actuelles pour maximiser sa récompense.

Il existe deux grandes classifications dans le RL : Model-based et Model-free RL. Les algorithmes de RL basés sur des modèles cherchent à l'apprentissage de la meilleure politique qui mène au but comme Policy Evaluation ou PEGASUS(Policy Evaluation-of Goodness And Search Using Scenarios) (NG & JORDAN, 2013). D'autre part, les algorithmes RL sans modèle cherchent à maximiser la récompense globale qui, bien sûr, mène au but, notamment Q-Learning et Temporal Difference (TD) (RAVISHANKAR & VIJAYAKUMAR, 2017).

— **Q-Learning**

Le Q-learning est un algorithme d'apprentissage par renforcement model-free. Il

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

utilise une fonction appelée fonction Q. Cette dernière estime la récompense future actualisée attendue pour une action particulière dans un état donné. Le Q-learning fonctionne en mettant à jour de manière itérative la fonction Q sur la base des récompenses observées et des transitions entre les états qui se produisent pendant les interactions de l'agent avec l'environnement. À chaque pas de temps, l'agent observe l'état actuel de l'environnement, sélectionne une action en fonction de la fonction Q, et reçoit une récompense et un nouvel état. La fonction Q est ensuite mise à jour sur la base de la récompense observée et des récompenses futures estimées du nouvel état et des actions disponibles (GÉRON, 2022).

La mise à jour de la fonction Q est faite ainsi (KAELBLING et al., 1996) :

$$Q(s, a) = Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)) \quad (1.1)$$

$Q(s, a)$ est l'estimation actuelle de la récompense future actualisée attendue pour l'action a dans l'état s . α est le taux d'apprentissage, qui contrôle le poids accordé aux nouvelles observations par rapport aux estimations précédentes. r est la récompense reçue après avoir effectué l'action a dans l'état s et être passé à l'état s' . γ est le facteur d'actualisation, qui détermine le poids accordé aux récompenses futures par rapport aux récompenses immédiates, et $\max_{a'} Q(s', a')$ est la valeur Q maximale pour les actions disponibles dans le nouvel état s' .

- **Temporal Difference Learning** Contrairement à l'apprentissage Q, qui apprend une politique optimale directement en estimant la valeur de chaque paire état-action, le TD-Learning est une approche model-free qui met à jour son estimation de la récompense attendue dans chaque état en fonction de la récompense observée et de la valeur prédictive de l'état suivant. Le TD-Learning met à jour la fonction de valeur de manière incrémentale en comparant la valeur prédictive de l'état actuel à la récompense observée et à la valeur prédictive de l'état suivant. Plus précisément, à chaque pas de temps t , l'algorithme d'apprentissage TD met à jour la fonction de valeur $V(s_t)$ pour l'état actuel s_t comme suit :

$$V(s_t) = V(s_t) + \alpha \cdot (r_t + 1 + \gamma V(s_t + 1) - V(s_t)) \quad (1.2)$$

α est le taux d'apprentissage, qui contrôle le poids donné aux nouvelles observations par rapport aux estimations précédentes. $r_t + 1$ est la récompense reçue après avoir effectué une action dans l'état s_t et être passé à l'état $s_t + 1$. γ est le facteur d'actualisation, qui détermine le poids accordé aux récompenses futures par rapport aux récompenses immédiates, et $V(s_t + 1)$ est la valeur prédictive de l'état suivant.

- **Deep Reinforcement Learning**

Le (DRL) est une branche de l'apprentissage automatique qui combine des techniques d'apprentissage profond avec l'apprentissage par renforcement. Il utilise des réseaux neuronaux pour représenter la politique ou la fonction de valeur d'un agent dans un espace d'état à hautes dimensions. Cela permet à l'agent d'apprendre des politiques plus complexes et sophistiquées qui peuvent résoudre des problèmes plus difficiles que les algorithmes d'apprentissage par renforcement traditionnels (ARULKUMARAN et al., 2017 ; GÉRON, 2022).

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

L'algorithme DRL le plus courant est appelé Deep Q-Networks (DQN). Il s'agit d'une extension de l'apprentissage Q qui utilise des réseaux de neurones profonds pour représenter la fonction Q. L'algorithme DQN utilise une technique appelée relecture d'expérience, qui stocke les expériences de l'agent dans une mémoire tampon et les échantillonne de manière aléatoire afin de rompre toute corrélation entre les expériences. Cela améliore la stabilité du processus d'apprentissage et permet à l'agent d'apprendre à partir d'événements rares qui peuvent ne pas se produire fréquemment (ARULKUMARAN et al., 2017).

Autres catégories

En plus des types de ML que nous venons de citer, il y a aussi des sous catégories ou une autre classification du ML comme les réseaux de neurones qui peuvent être supervisés, non supervisés ou par renforcement, l'apprentissage semi-supervisé, ensembliste, multitâches et celui basé sur les instances. Ces différentes catégories seront élaborées en annexe.

1.3.3 Fonctionnement

Un modèle générique de l'apprentissage automatique se compose de certaines étapes qui sont indépendantes de l'algorithme d'apprentissage choisi (ALZUBI et al., 2018). Dans ce qui suit, nous allons voir ce processus d'apprentissage automatique.

Processus de l'apprentissage automatique

Le processus d'apprentissage automatique consiste à fournir à un algorithme ML des données d'apprentissage pour ensuite générer un modèle qui sera utilisé pour prédire des résultats. Cela comprend la collecte des données, leur pré-traitement, le choix d'un algorithme approprié, la sélection du modèle et des paramètres, l'entraînement du modèle sur les données, son évaluation et enfin son déploiement (ALZUBI et al., 2018 ; « Introduction to Machine Learning », 2022 ; MAYO, 2022) :

1. **Collection et préparation des données** : La première étape est de collecter les données qui seront utilisées dans l'apprentissage. Ces données peuvent provenir de diverses sources telles que des capteurs, des enquêtes, des réseaux sociaux et des bases de données. Une fois les données collectées, elles doivent être nettoyées, traitées et formatées afin de pouvoir être utilisées. Cela implique des tâches telles que la suppression des doublons, le traitement des valeurs manquantes et la transformation des données dans un format utilisable par l'algorithme d'apprentissage automatique. Par la suite, ces données traitées seront divisées sur deux datasets, le premier sera utilisé pour l'entraînement du modèle et le second sera utilisé pour le test et l'évaluation.
2. **Sélection de l'algorithme ML** : Tous les algorithmes d'apprentissage automatique ne sont pas destinés à tous les problèmes. Pour chaque type de problème il existe des algorithmes appropriés. La sélection du meilleur algorithme pour le problème à résoudre est impérative pour obtenir les meilleurs résultats possibles.

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

3. **Sélection du modèle et des paramètres :** La plupart des algorithmes ML nécessitent des valeurs initiales pour les différents paramètres. Ces valeurs sont à définir manuellement. Ce choix a un grand impact sur la performance du modèle.
4. **Entraînement du modèle :** L'entraînement d'un modèle d'apprentissage automatique consiste à ajuster les paramètres internes du modèle afin de minimiser la différence entre ses prédictions et les résultats réels des données d'entraînement. Cette étape dépend du type d'apprentissage choisi. Voici plus en détails les étapes de l'entraînement d'un modèle ML dans le cas le plus simple, un apprentissage supervisé :
 - (a) Initialisation des paramètres internes du modèle.
 - (b) Entraînement : Une fois les paramètres du modèle initialisés, les données d'apprentissage sont introduites dans le modèle, un exemple à la fois. Pour chaque exemple, il calcule une prédition basée sur les données d'entrée et ses paramètres internes actuels.
 - (c) Calcul de la perte/coût : Après que le modèle ait effectué une prédition pour un exemple d'entrée, la différence entre la sortie prédite et la sortie réelle est calculée, en utilisant une fonction de perte (appelée aussi fonction de coût). Cette différence permet de mesurer la performance du modèle sur les données d'apprentissage.
 - (d) Itération : Le processus consistant à introduire un exemple dans le modèle, à calculer la perte et à mettre à jour les paramètres est répété pour tous les exemples des données d'apprentissage. Chaque passage à travers l'ensemble des données d'apprentissage est appelé une epoch. Les paramètres internes du modèle sont ajustés à la fin de chaque epoch pour améliorer ses performances sur les données d'apprentissage.
 - (e) Critères d'arrêt : L'entraînement du modèle se poursuit jusqu'à ce qu'un critère d'arrêt soit rempli, il peut être un nombre fixe d'epoch, un niveau de performance cible ou autres conditions.
5. **Évaluation du modèle :** Après l'entraînement du modèle, il faut évaluer ses performances avec les données de test. Cela donnera une idée de la capacité du modèle à faire des prédictions sur de nouvelles données.
6. **Ajustement du modèle :** Si les performances du modèle ne sont pas satisfaisantes, un ajustement des hyperparamètres doit être fait et ensuite toutes les étapes de l'entraînement sont répétées jusqu'à l'obtention de résultats satisfaisants.
7. **Déploiement du modèle :** Une fois la performance voulue est obtenue, le modèle peut être utilisé pour prédire les sorties des nouvelles données et des exemples non étiquetés.

1.3.4 Défis

Les deux composantes principales de l'apprentissage automatique sont les données et le modèle construit. Les défis auxquels le ML peut être confronté sont liés soit à de

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

mauvaises données, soit à un mauvais modèle. Dans ce qui suit, nous allons voir quelques challenges et leurs impactes sur le ML (GÉRON, 2022) :

- **Quantité insuffisante de données** : Plus le nombre de données impliquées dans l'apprentissage est important, meilleures sont les performances du modèle, quel que soit le type d'algorithme ML utilisé.
- **Données d'entraînement non-représentatives** : L'ensemble d'entraînement doit être le plus représentatif possible et il doit inclure tous les différents cas et classes. Si l'ensemble est très petit ou la méthode d'échantillonnage est défectiveuse, l'ensemble des données va contenir des échantillons bruits, et cela s'appelle le biais d'échantillonnage.
- **Mauvaise qualité de données** : Les données de mauvaise qualité signifient qu'elles contiennent des erreurs, des bruits, des valeurs aberrantes ou que certaines instances sont dépourvues de quelques caractéristiques, et tous ces éléments détériorent la qualité de l'apprentissage.
- **Caractéristiques non pertinentes** : La sélection des caractéristiques est une étape crucial dans l'apprentissage automatique, car elle permet au système de mieux apprendre si les caractéristiques sont représentatives et détiennent des informations pertinentes.
- **Sur-adaptation des données d'entraînement** : Il s'agit du cas où un modèle devient trop complexe et apprend à s'adapter de façon excessive aux données d'apprentissage, au détriment de sa capacité à se généraliser à de nouvelles données non vues. Parmi les techniques possibles pour résoudre ce problème : la simplification du modèle soit en diminuant les caractéristiques ou en choisissant une méthode plus simple, l'utilisation de plus de données variées d'entraînement, ou bien la réduction des bruits dans l'échantillon utilisé.
- **Sous-adaptation des données d'entraînement** : Il s'agit du cas où un modèle est trop simple et ne parvient pas à capturer les liens et patterns dans les données. Il y a sous-adaptation lorsque le modèle n'est pas en mesure d'apprendre les caractéristiques ou les relations pertinentes dans les données et que ses performances sont médiocres tant sur les données d'apprentissage que sur les données de test. Pour résoudre ce problème, les solutions suivantes permettent d'y remédier : choisir un modèle plus complexe et performant, choisir des caractéristiques de meilleur qualité, ou bien réduire les contraintes du modèle.

1.4 Conclusion

Dans ce chapitre nous avons défini ce qu'était l'IoT et les challenges auxquels il fait face. Nous savons maintenant ce qu'est l'apprentissage automatique et comment il exploite les données à l'aide de techniques de science des données. Pour finir, nous avons passé en revue certains types d'algorithmes d'apprentissage automatique.

Avec la vitesse à laquelle ces technologies progressent, le besoin d'apprentissage automatique augmente considérablement. En revanche, la nature contraignante et très distribuée de l'IoT ainsi que la sensibilité des données générées, nécessitent des méthodes

CHAPITRE 1. INTERNET DES OBJETS ET APPRENTISSAGE AUTOMATIQUE

d'apprentissage automatique améliorées et plus avancées, qui prennent en considération les contraintes imposées dans le domaine des IoT. Ces considérations, spécifiquement ceux liés à la vie privée, empêchent les données d'être apportées à un dépôt de données central pour les utiliser dans le processus de ML.

L'apprentissage fédéré (FL) est la solution à cette problématique, c'est une approche qui permet d'entraîner des modèles ML sur des données situées dans des endroits disparates, ne nécessitant ni le partage ni la collecte centrale des données. Cependant, il apporte avec lui son lot d'inconvénients en particulier les risques liés à la sécurité et la disponibilité du serveur central. La blockchain se trouve être une remède à ces vulnérabilités. Dans le chapitre suivant, nous allons voir plus en détails ces technologies et comment elles apportent des éléments de solution à la problématique citée.

Chapitre 2

Apprentissage Fédéré et Blockchain

2.1 Introduction

L'IoT fait partie des sources les plus importantes de génération de données, la science des données est là pour apporter une contribution considérable afin de rendre les applications IoT plus intelligentes (MAHDAVINEJAD et al., 2018). En vue de la nature distribuée des systèmes IoT, des techniques plus adaptées ont été proposées pour résoudre les défis d'hétérogénéité des dispositifs ainsi que la sensibilité des données traitées et des communications. L'apprentissage fédéré est une nouvelle méthode d'apprentissage (C. ZHANG et al., 2021), qui permet non seulement de surmonter ces challenges, mais aussi d'aider à atteindre un bien meilleur niveau d'évolutivité et de préservation de la confidentialité, et cela à travers une coordination de plusieurs dispositifs pour effectuer l'apprentissage de l'IA à la périphérie des réseaux IoT tout en conservant les données en sécurité sur les dispositifs locaux.

Dans ce chapitre, nous allons présenter l'apprentissage fédéré et comment il a réussi à régler le problème d'hétérogénéité et de distributivité. Nous parlerons aussi de l'intégration du FL dans l'IoT et des différentes architectures et algorithmes déjà existants. En outre, nous présenterons la blockchain et son rôle dans la décentralisation du processus d'apprentissage et la sécurisation de l'environnement.

2.2 Apprentissage Fédéré

2.2.1 Définition

De plus en plus de gens sont méfiants à l'égard de l'intelligence artificielle, en ce qui concerne leur vie privée et la confidentialité de leurs données, notamment les gouvernements, à tel point que de nouvelles lois et règles ont vu le jour. Citons par exemple les politiques telles que le règlement général sur la protection des données (General Data Protection Regulation GDPR), implanté en mai 2018 par l'UE, qui stipule des règles sur le partage des données entre différentes organisations (ALBRECHT, 2016), ou bien le CCPA aux USA (MACTAGGART & MACTAGGART, 2021) ou le PDPA à Singapour (CHIK, 2013). Malgré cela, les entreprises doivent relever de grands défis pour protéger la vie privée de

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

leurs clients et maintenir l'intégrité des données. Même des titans comme Google peuvent trébucher : L'entreprise a écoper d'une amende de 57 millions d'euros pour une faille de GDPR.

Le traitement des données générées par l'IoT et l'application d'algorithmes ML devient dur à mettre en œuvre, et ce en raison de la nature contraignante des dispositifs IoT de faible capacité de calcul et de nature hétérogène. Les lois et règlements sur la protection de la confidentialité des données ne facilitent pas la tâche. Mais encore, la méthode d'entraînement centralisée peut entraîner une fuite de données et une atteinte à la vie privée du propriétaire des données (C. ZHANG et al., 2021). Elle représente un point faible de l'intégration du ML aux IoT. Avec l'aide de l'apprentissage fédéré, les utilisateurs individuels peuvent bénéficier de l'obtention d'un modèle d'apprentissage automatique performant sans avoir à transmettre leurs données personnelles sensibles à un serveur central.

Le FL permet à plusieurs parties de former conjointement un modèle d'apprentissage automatique sans échanger leurs données locales. Elle couvre les techniques de plusieurs domaines de recherche tels que les systèmes distribués, l'apprentissage automatique et la confidentialité. Dans un système d'apprentissage fédéré, plusieurs parties forment en collaboration des modèles d'apprentissage sans échanger leurs données brutes, ces modèles seront ensuite agrégés par un serveur central (Q. LI et al., 2021). Cette approche s'oppose aux méthodes traditionnelles d'apprentissage automatique centralisées, où tous les ensembles de données locales sont téléchargés vers un serveur. Elle s'oppose aussi aux approches décentralisées plus classiques qui supposent souvent que les échantillons de données locales sont distribués de manière identique. L'apprentissage fédéré permet à de multiples acteurs de construire un modèle ML commun et robuste sans partager leurs données, ce qui permet de traiter des enjeux critiques tels que la confidentialité des données, la sécurité, les droits d'accès et l'accès à des données hétérogènes (NGUYEN et al., 2021).

2.2.2 Fonctionnement de l'apprentissage fédéré

L'apprentissage fédéré est une approche distribuée de l'apprentissage automatique qui permet de faire un entraînement des modèles sur des sources de données décentralisées tout en préservant la confidentialité des données. Les principales étapes de l'apprentissage fédéré sont les suivantes (NIKNAM et al., 2020) :

1. **Initialisation** : Un serveur central initialise un modèle global, qui est ensuite envoyé aux clients participants.
2. **Formation locale** : Chaque client entraîne le modèle global sur ses données locales, en utilisant ses propres ressources informatiques. Le processus de formation peut être personnalisé en fonction des besoins spécifiques du client, ce qui permet une flexibilité et une adaptation à diverses distributions et caractéristiques de données.
3. **Agrégation du modèle** : Une fois l'apprentissage local terminé, les clients envoient les paramètres actualisés de leur modèle (gradients ou poids) au serveur central, qui agrège les paramètres des modèles pour créer un nouveau modèle global. L'agrégation peut être réalisée à l'aide de différentes méthodes, telles que la moyenne ou la

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

moyenne pondérée, en fonction des exigences spécifiques de l'application.

4. **Processus itératif** : Les étapes 2 et 3 sont répétées pendant un certain nombre d'itérations ou jusqu'à ce qu'un critère de convergence soit atteint. Ce processus itératif permet au modèle global de s'améliorer au fil du temps en intégrant les connaissances des modèles locaux.

2.2.3 Types d'apprentissage fédéré

Dans la littérature, nous retrouvons plusieurs classifications du FL. D'après (Q. LI et al., 2021 ; C. ZHANG et al., 2021) les techniques FL peuvent être catégorisées selon les aspects représentés dans la figure 2.1 :

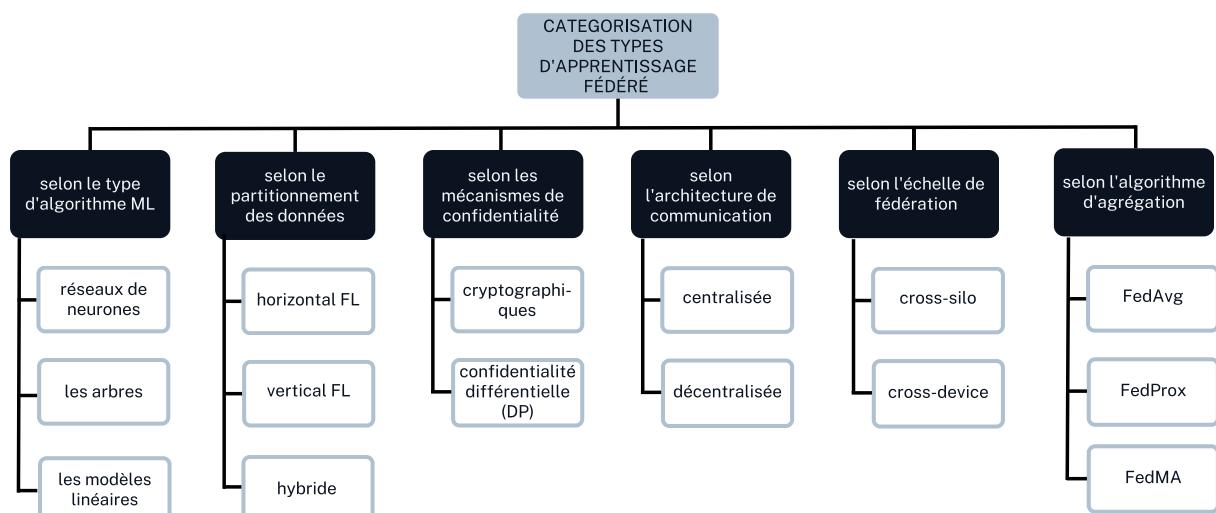


FIGURE 2.1 – Les différentes catégorisation d'apprentissage fédéré (Q. LI et al., 2021 ; C. ZHANG et al., 2021).

Les modèles de machine learning

L'apprentissage fédéré peut utiliser un large éventail d'algorithmes d'apprentissage automatique pour l'entraînement locale, en fonction de la tâche spécifique et des sources de données concernées. Les plus couramment utilisés sont les réseaux neuronaux profonds (DNN), les arbres de décision, les forêts aléatoire, SVM et la régression logistique (LUDWIG & BARACALDO, 2022)

La répartition des données

Basé sur la façon dont les données sont distribuées dans l'espace des échantillons et des caractéristiques, nous distinguons trois types de FL, tels que présentés dans la figure 2.2 (NGUYEN et al., 2021).

- **L'apprentissage fédéré horizontal (HFL)** : Dans l'apprentissage fédéré horizontal, plusieurs clients forment conjointement un modèle sous la coordination du

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

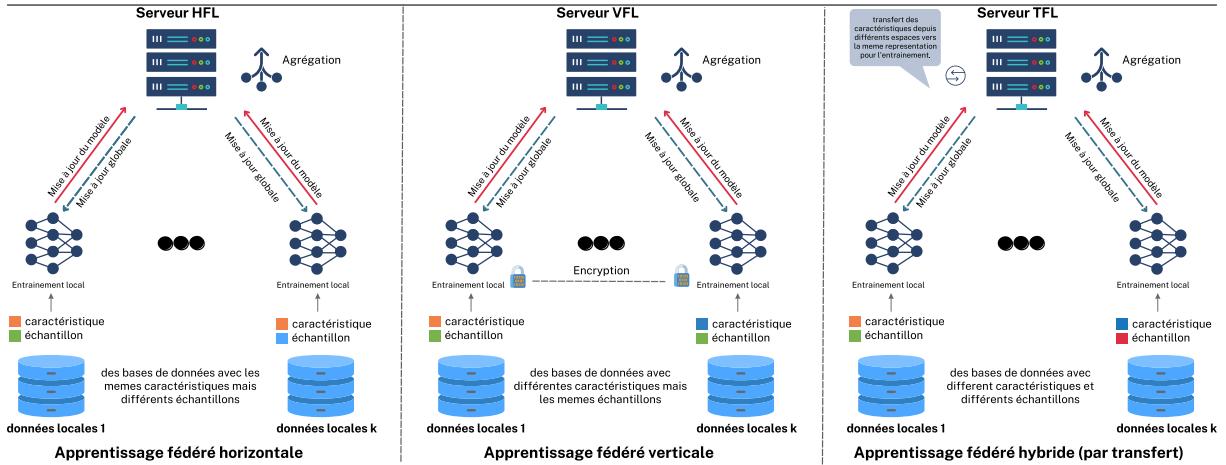


FIGURE 2.2 – Les types d'apprentissage fédéré selon le partitionnement des données (NGUYEN et al., 2021).

serveur central. Les données d'apprentissage sont partitionnées horizontalement, ce qui signifie que chaque client dispose d'un sous-ensemble différent d'échantillons mais que tous partagent le même espace de caractéristiques. Le serveur central agrège les modèles formés par chaque client et envoie un modèle mis à jour à tous les clients pour une formation supplémentaire. Ce processus se répète jusqu'à ce que la convergence ou un autre critère d'arrêt soit atteint (HUANG et al., 2021). Dans l'apprentissage fédéré horizontal continu (CHFL), une approche d'apprentissage continu est utilisée pour mettre à jour le modèle au fil du temps, à mesure que de nouvelles données soient disponibles(MORI et al., 2022). Dans (HUANG et al., 2021), ils ont proposé FedFa, un algorithme pour atteindre plus d'équité et de précision dans l'apprentissage fédéré horizontal.

- **L'apprentissage fédéré vertical (VFL) :** il est basé sur les caractéristiques, les ensembles de données des différentes parties ont un espace d'échantillonnage identique ou similaire, mais diffèrent dans l'espace des caractéristiques. Afin de collecter et regrouper les différentes caractéristiques il se repose sur les techniques d'alignement des entités. Un exemple de cas d'utilisation du FL vertical est lorsque des organisations collaboratrices possèdent des données du même ensemble de clients mais ont des espaces d'attributs différents (A. KHAN et al., 2022).
- **L'apprentissage fédéré hybride (TFL) :** aussi appelé apprentissage fédéré par transfert, permet de transférer des connaissances entre domaines. Il s'agit d'un schéma d'apprentissage où les connaissances sont transférées d'une partie à une autre, qui ne dispose pas de suffisamment de données. Le FTL peut être utilisé pour apprendre à partir de données décentralisées et transférer des informations d'un domaine à un autre. Ce type d'apprentissage peut être utilisé pour pallier le manque de données ou bien d'étiquettes (NGUYEN et al., 2021).

Les mécanismes de confidentialité

Même si les données ne sont pas exposées aux risques de confidentialité, les paramètres et résultats de l'apprentissage, eux, le sont et peuvent révéler plus d'informations qu'on ne pourrait le croire. Afin de pallier à cela le FL a adopté différents mécanismes de confidentialité, qui sont catégorisés en deux grandes approches :

- **Les méthodes cryptographiques** : on cite notamment le chiffrement homomorphe (*homomorphic encryption*) et le calcul multipartite sécurisé (*secure multi-party computation SMC*)
- **La confidentialité différentielle** : La confidentialité différentielle ajoute du bruit aux données de manière contrôlée tout en permettant d'extraire des informations précieuses des ensembles de données. L'une des principales caractéristiques de la confidentialité différentielle est qu'elle fournit une garantie qui tient, indépendamment de ce que sait ou fait un adversaire lorsqu'il effectue des attaques sur les données. Un adversaire est quelqu'un qui essaie d'apprendre des informations sensibles sur des individus à partir des données. Même si de nouvelles informations supplémentaires sont disponibles, la confidentialité différentielle offre toujours exactement le même niveau de protection, ce qui en fait un outil efficace pour garantir la sécurité des données privées des individus (UL HASSAN et al., 2018).

L'architecture de la communication

- **Architecture centralisée** : dans la conception centralisée, le flux de données est souvent asymétrique, ce qui signifie que le gestionnaire agrège les informations (par exemple, les modèles locaux) des parties et renvoie les résultats de l'entraînement. Les mises à jour des paramètres du modèle global sont toujours effectuées par ce gestionnaire. La communication entre le gestionnaire et les autres parties peut être synchrone ou asynchrone.
- **Architecture décentralisée** : dans une conception décentralisée, les communications sont effectuées entre les parties (pair à pair). Chaque partie est capable de mettre à jour les paramètres globaux directement, tel qu'illustré dans la figure 2.3 (NGUYEN et al., 2021).

L'échelle de fédération

Les systèmes FL peuvent être classées en deux types selon l'échelle de la fédération : les systèmes intersilo et inter-appareils. Les différences entre eux résident dans le nombre de parties et la quantité de données stockées dans chaque partie.

- **Cross-silo** : parfois référencé comme systèmes FL privés, il se caractérise par peu d'entités mais chacune avec beaucoup de données et de capacités de calcul.
- **Cross-device** : ou systèmes FL publics ; un grand nombre d'entités, mais chacune avec peu de données et capacités de calcul.

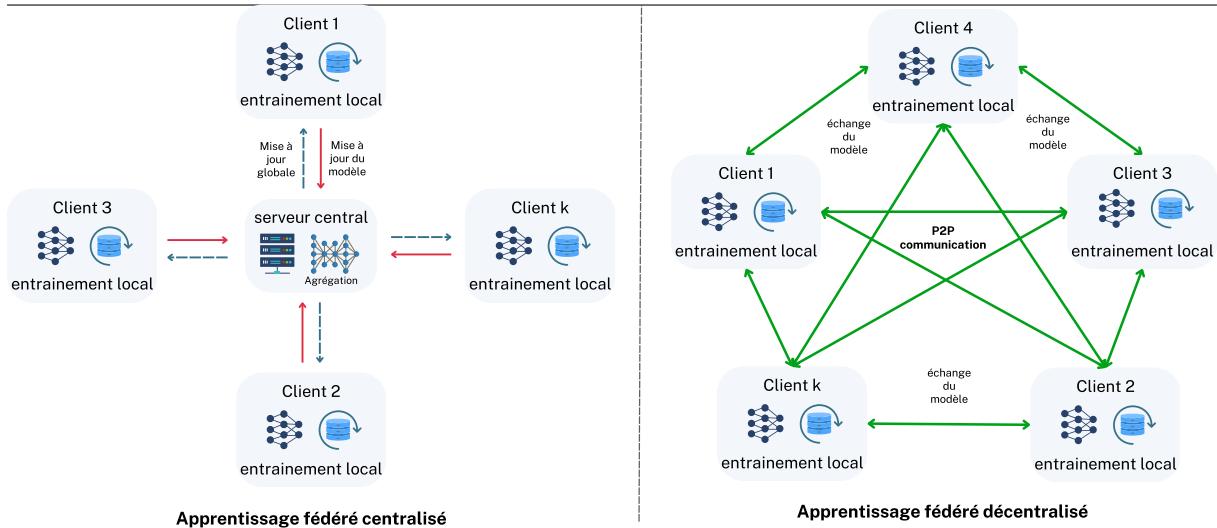


FIGURE 2.3 – Les types d'apprentissage fédéré selon l'architecture de communication (NGUYEN et al., 2021).

L'algorithme d'agrégation

Les algorithmes d'agrégation sont multiples, nous allons lister quelques uns :

- **FedAvg** : c'est l'algorithme le plus utilisé dans l'apprentissage fédéré. Il permet aux clients de conserver leurs données localement et un serveur central de paramètres est utilisé pour communiquer entre eux. Ce serveur central distribue les paramètres à chaque client et collecte les paramètres mis à jour des clients. FedAvg est un algorithme efficace en termes de communication pour la formation distribuée avec un nombre énorme de clients. En pratique, la moyenne naïve des paramètres fonctionne de façon surprenante dans FedAvg. Cependant, dans un environnement hétérogène ses performances se dégradent (RJOURB et al., 2022).
- **FedProx** : c'est un framework pour l'optimisation fédérée dans les réseaux hétérogènes. Il s'agit d'une généralisation de FedAvg qui prend en compte l'hétérogénéité des données et des systèmes en repartageant la fonction objectif. L'apprentissage est effectué par cycles, dans lesquels le serveur échantillonne un ensemble de clients et leur envoie le modèle global actuel (MOTHUKURI et al., 2021 ; RJOURB et al., 2022).
- **FedMA (FL with Matched Averaging)** : c'est un algorithme conçu pour l'apprentissage fédéré d'architectures de réseaux neuronaux modernes tel que les réseaux neuronaux convolutifs (CNN). FedMA construit le modèle global partagé par couches en faisant correspondre et en moyennant des éléments cachés avec des signatures d'extraction de caractéristiques similaires. Les expériences indiquent que FedMA surpassé les algorithmes d'apprentissage fédéré les plus populaires sur les architectures CNN profondes entraînées sur des ensembles de données du monde réel, tout en réduisant la charge de communication globale (H. WANG et al., 2020).

2.2.4 Défis

Afin de protéger efficacement la vie privée des entreprises et des utilisateurs, certains défis doivent être résolus dans l'apprentissage fédéré, parmi lesquels nous citons :

Fuite des données (information leakage)

La fuite des données arrive lorsqu'il y a un échange des paramètres du modèle dans un réseau : certaines informations concernant la vie privée des participants peuvent être déduites à partir des mise à jour des modèles. Puisque l'objectif principal de l'apprentissage fédéré est de résoudre le problème de la protection des données privées dans l'apprentissage automatique, le modèle d'apprentissage dans le FL ne doit pas révéler les informations privées des utilisateurs. Il existe des solutions à ce problème tels que la confidentialité différentielle (DP) ou bien l'homomorphic encryption (HE), cependant ces méthodes dégradent les performances du modèle global et élèvent le coût de calcul et de communication (ISSA et al., 2023 ; WEI et al., 2020).

Hétérogénéité

Il existe un grand nombre de dispositifs edge ou appareils IoT dans les environnements d'apprentissage fédéré, ce type d'appareils sont très hétérogènes sur différents niveaux : la capacité de calcul, la consommation d'énergie, les protocoles de communication ou bien leur système d'exploitation (NISHIO & YONETANI, 2018). Les données détenues par ces dispositifs peuvent être non IID (Non-Independent and Identically Distributed). Par exemple, dans un système médical intelligent, la structure des données des dossiers médicaux électroniques de différents types de maladies est différente. Ce qui représente un grand défi pour entraîner ces données non IID (L. U. KHAN et al., 2021).

Sécurité

La sécurité est un défi important pour l'apprentissage fédéré car les données sont distribuées sur plusieurs appareils, ce qui les rend vulnérables aux menaces de sécurité. Les attaques par empoisonnement des données/modèles et les attaques par inférence sont les principaux risques pour la sécurité de l'apprentissage fédéré. Le processus de synchronisation est également vulnérable aux attaques de type "man-in-the-middle". De plus, garantir l'intégrité et l'authenticité des mises à jour du modèle est un défi important. Des techniques telles que l'agrégation sécurisée, le cryptage homomorphique et la confidentialité différentielle, ont été proposées pour relever ces défis. Toutefois, la mise en œuvre de ces techniques peut s'avérer difficile en raison de leur complexité de calcul, de leur surcharge de communication et de la nécessité de disposer de matériel ou de logiciels spécialisés (ISSA et al., 2023 ; MOTHUKURI et al., 2021 ; K. ZHANG et al., 2022).

Sélection des participants

La sélection des participants est une tâche délicate dans l'apprentissage fédéré en raison de l'hétérogénéité des nœuds participants, des problèmes de confidentialité et de

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

sécurité, des coûts de communication, de la nature dynamique du système et de l'introduction potentielle de biais dans le processus d'apprentissage. De plus les nœuds peuvent être corrompus ou malveillants : ils peuvent attaquer le système par exemple à travers des attaques d'empoisonnement. La sélection des bons nœuds est essentielle pour assurer une entraînement efficace des modèles dans l'apprentissage fédéré (L. U. KHAN et al., 2021 ; NGUYEN et al., 2021).

Synchronisation

La synchronisation est un défi pour l'apprentissage fédéré car dans un environnement synchrone, le serveur doit attendre les mises à jour du modèle de tous les clients participants à la tâche. Cela peut être inefficace et ralentir le processus (WEN et al., 2022). Plusieurs solutions à ce problème ont été proposées, parmi lesquelles nous citons : la synchronisation partielle (QU et al., 2022), ou bien la planification des participants (client scheduling) (NISHIO & YONETANI, 2018).

Cependant, malgré tous ces challenges, l'apprentissage fédéré a réussi à valoriser ses nombreux avantages et à se positionner dans le contexte des nouvelles technologies telles que les villes et les industries intelligentes.

2.2.5 Application de l'apprentissage fédéré pour IoT

L'apprentissage fédéré est très pertinent pour les IoT. En permettant un apprentissage sans partage de données, il préserve leur confidentialité, il réduit la latence de communication et diminue les ressources réseau utilisées. De plus, il améliore la qualité d'apprentissage, grâce à l'implication de multiples datasets et la non limitation de ressources de calculs (i.e le nombre de participant dans le processus d'apprentissage n'est pas limité, par conséquent le modèle converge plus rapidement et donne une meilleure précision)(NGUYEN et al., 2021). Le FL a permis la mise en place de systèmes IoT intelligents qui ont fait évoluer de nombreux domaines tels que (NGUYEN et al., 2021) :

Smart Healthcare

L'utilisation de l'apprentissage fédéré dans les soins de santé intelligents a plusieurs application dont la gestion des dossiers médicaux électroniques (DME) et de l'imagerie médicale. Le FL aide les hôpitaux à analyser les données des DME et à surveiller la santé des patients, ainsi qu'à créer des dispositifs IoT intelligents qui permettent aux médecins d'accéder aux données de plusieurs établissements médicaux sans compromettre la confidentialité des patients. En outre, il permet aux établissements de santé de partager des données et d'entraîner ensemble un modèle d'apprentissage automatique pour améliorer le diagnostic des maladies, la découverte de médicaments et la planification des traitements (DASARADHARAMI REDDY & GADEKALLU, 2023).

Smart City

L'apprentissage fédéré a de nombreuses applications dans le domaine de l'IoT pour les villes intelligentes, notamment les réseaux intelligents (Smart Grid) et la gestion intelligente des données(Smart Data Management). Dans le contexte d'un réseau intelligent, le FL peut être utilisé pour gérer la demande d'énergie de différents consommateurs en prédisant leurs habitudes de consommation d'énergie. Cela peut aider les services publics à mieux gérer leur approvisionnement en énergie et à réduire le gaspillage. Le FL peut également être utilisé dans la gestion intelligente des données pour permettre le partage sécurisé et respectueux de la vie privée des données entre plusieurs organisations.Les villes peuvent ainsi optimiser leurs ressources et leurs services, ce qui conduit à des environnements urbains plus durables et plus efficaces (NGUYEN et al., 2021).

Smart Industry

Dans le contexte de la robotique et de l'industrie 4.0, le FL peut permettre un apprentissage collaboratif entre les robots sans partage des données. Cette approche peut améliorer l'efficacité et la précision des tâches robotiques tout en protégeant les données sensibles. De plus, le FL peut être appliqué aux réseaux IoT edge-based, où les données sont traitées et analysées au niveau edge du réseau plutôt que d'être envoyées à un serveur central. Cette approche peut réduire la latence et améliorer les capacités de prise de décision en temps réel. Il peut également être utilisé dans les bancs d'essai pour l'IoT industriel, où plusieurs entreprises peuvent collaborer pour tester et développer de nouvelles applications et services IoT. Cette approche peut promouvoir l'innovation et accélérer l'adoption de l'IoT dans les industries (NGUYEN et al., 2021).

Smart Transportation

Le FL peut être utilisé pour gérer la planification du trafic des véhicules et la gestion des ressources des véhicules (NGUYEN et al., 2021). Dans la planification du trafic, FL peut aider à optimiser le flux et à réduire la congestion en analysant les données de trafic en temps réel. Dans la gestion des ressources, FL peut aider à gérer les véhicules d'une flotte en optimisant les itinéraires et les horaires, ainsi qu'en surveillant leurs performances et leurs besoins de maintenance. Cela peut conduire à des systèmes de transport plus efficaces et plus rentables. Dans l'ensemble, le FL dans les transports intelligents peut contribuer à améliorer la fluidité du trafic, à réduire les émissions et à améliorer la mobilité globale (RAMU et al., 2022).

L'objectif principal de l'apprentissage fédéré est de répondre aux préoccupations de confidentialité et de sécurité des clients, en conservant leurs données localement. Cependant, l'étape d'agrégation dans le FL nécessite un serveur central, ce qui représente une énorme vulnérabilité car il s'agit d'un point de défaillance unique qui peut être la cible de nombreuses menaces, telles que des attaques par empoisonnement ou des goulets d'étranglement dans les communications (MOTHUKURI et al., 2021). D'où la nécessité d'un élément décentralisé, qui permettra d'établir une confiance entre les participants et une transparence dans le processus d'apprentissage, en particulier dans les environnements

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

IoT, qui parfois sont très critiques tels que les applications de smart healthcare (ALSHEHRI & MUHAMMAD, 2020) ou bien les applications du gouvernement (KANKANHALLI et al., 2019). Dans des études récentes (OTOUM et al., 2022 ; SHARMA et al., 2022 ; ZHAO et al., 2020), une nouvelle technologie, la blockchain, est explorée afin de résoudre ces préoccupations de confidentialité et sécurité dans les systèmes d'apprentissage fédéré. Telle qu'elle a été décrite dans (C. MA et al., 2022) : " En tant que technologie sécurisée, la blockchain a la capacité de tolérer un point de défaillance unique grâce au consensus distribué, et elle peut en outre mettre en œuvre des mécanismes d'incitation pour encourager les participants à contribuer efficacement au système.". Dans ce qui suit nous allons définir cette nouvelle technologie et voir comment l'IoT peut en bénéficier.

2.3 Blockchain

2.3.1 Définition

Le concept de blockchain est apparu pour la première fois en 2008, proposé par Nakamoto pour le Bitcoin (NAKAMOTO, 2008). Depuis, la blockchain constitue le pilier du système moderne de crypto-monnaie numérique. Ce réseau de blocs est décentralisé par nature, et cette propriété permet de stocker la monnaie de manière plus sûre et d'introduire la propriété de transparence (ALI et al., 2021). Cette propriété unique de la gestion de la monnaie virtuelle par les nœuds du réseau attire l'attention des chercheurs et de ceux qui découvrent la technologie blockchain dans les moindres détails pour trouver plusieurs applications de cette technologie dans d'autres domaines émergents tels que la finance, les soins de santé, l'enregistrement des actifs et de nombreux autres domaines.

La blockchain est une base de données distribuée ou un grand registre qui stocke des données dans des structures appelées blocs. La mise à jour d'une blockchain est répartie entre les nœuds ou les participants d'un réseau public ou privé régi par des règles convenues par les participants du réseau, ce qui est connu sous le nom de technologie du grand registre distribué (distributed ledger technology) (HABIB et al., 2022).

Chaque bloc de la blockchain est comme un paquet de données inaltérable, contenant un timestamp, un ensemble de transactions et la valeur de hachage du bloc précédent (parent), ainsi qu'un nonce, qui est un nombre aléatoire permettant de vérifier le hachage, tel qu'illustré dans la figure 2.4. La blockchain est étendue par chaque bloc supplémentaire et représente donc un registre complet de l'historique des transactions. Ce concept garantit l'intégrité de l'ensemble de la blockchain jusqu'au premier bloc (genesis block). Les valeurs de hachage sont uniques et la fraude peut être efficacement évitée puisque la modification d'un bloc dans la chaîne change immédiatement la valeur de hachage correspondante. Si la majorité des nœuds du réseau s'accordent par un mécanisme de consensus sur la validité des transactions d'un bloc et sur la validité du bloc lui-même, le bloc peut être ajouté à la chaîne (NOFER et al., 2017).

Les transactions stockées sur une blockchain peuvent être plus que de simples en-

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

registrements de l'échange d'actifs. Les systèmes de blockchain émergents permettent également de stocker des programmes informatiques et de les exécuter dans le cadre de transactions sur le registre. Ces programmes sont souvent appelés "contrats intelligents", même s'ils ne sont généralement pas très intelligents et ne sont souvent pas liés à des contrats juridiques. Comme ils ont été définis dans (XU et al., 2019) "Les contrats intelligents sont des programmes déployés en tant que données dans le registre de la blockchain et exécutés dans des transactions sur la blockchain. Les contrats intelligents peuvent détenir et transférer des actifs numériques gérés par la blockchain et peuvent invoquer d'autres contrats intelligents stockés sur la blockchain. Le code des contrats intelligents est déterministe et immuable une fois déployé".

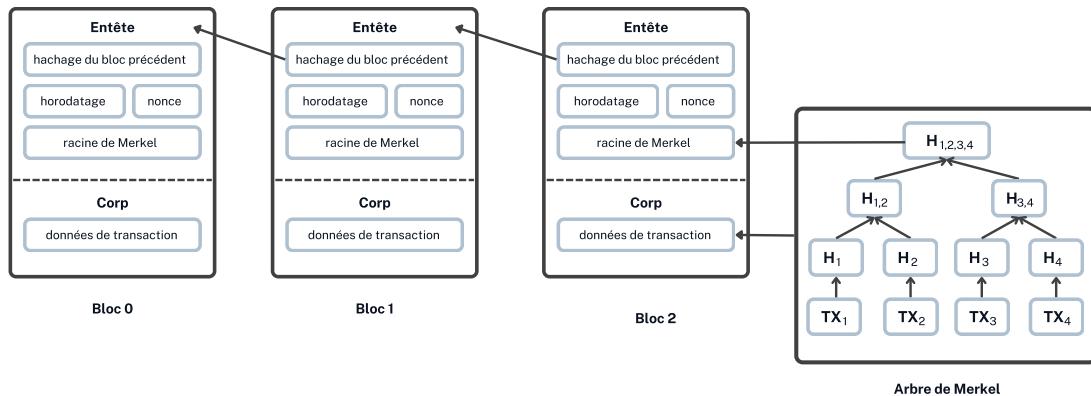


FIGURE 2.4 – La structure de la Blockchain (LIANG, 2020)

2.3.2 Types de Blockchain

Il existe trois types de Blockchain : publique, privée et consortium d'après (ALI et al., 2021) :

Blockchain publique

La blockchain publique peut être qualifiée de blockchain sans permission, car tout individu peut se joindre au réseau et exécuter le processus de transaction (DINH et al., 2018). Les noeuds mineurs recueillent les informations sur le bloc à partir du réseau et lancent le processus de minage une fois le processus de validation terminé. Les mineurs reçoivent ainsi certaines récompenses prédéfinies. Cependant, dans le cas d'un bloc public, l'identité de l'individu est inconnue et n'importe qui peut créer un bloc dans le réseau, ce qui rend la chaîne de blocs publics vulnérables aux cyberattaques. Ce mécanisme signifie que si une personne veut contrôler le réseau, elle doit avoir une approbation majoritaire de plus de 51 % du réseau et cela nécessite une grande puissance de frappe. En outre, pour sécuriser les opérations et les transactions entre les partenaires, la cryptographie à clé publique est utilisée. Cependant, la blockchain publique ne peut pas être utilisée dans certaines applications où la quantité de données est importante en raison de la complexité informatique élevée.

Blockchain privée

Le réseau de blockchain qui est établi au sein d'une organisation privée ou contrôlée par un groupe de personnes est appelé blockchain privée ou blockchain avec autorisation. La blockchain privée est de nature centralisée. Le processus d'extraction est contrôlé par l'organisation spécifique où la blockchain privée est introduite, ce qui rend plus difficile la création d'un bloc pour les externes, à moins que le personnel de la blockchain privée ne l'autorise (PUTHAL et al., 2018).

Blockchain consortium

Le réseau de blockchain qui est formé par la combinaison des blockchains publiques et privées est appelé blockchain de consortium ou bien blockchain par autorisation. Comme pour la blockchain privée, l'algorithme de validation et de consensus du bloc dépend entièrement d'un groupe ou d'une organisation (GU et al., 2018). Pour miner le bloc et l'ajouter au réseau de blockchain, le mécanisme de signature est adopté, ce qui signifie que le nouveau bloc ne sera ajouté que si les noeuds de contrôle l'ont approuvé en le signant. Cependant, les blocs ajoutés au réseau peuvent être tempérés facilement car la blockchain est contrôlée par un certain ensemble de noeuds.

2.3.3 Caractéristiques

Dans ce qui suit, une liste des caractéristiques les plus pertinentes est présentée :

- **Persistante** : les transactions enregistrées dans un registre blockchain sont considérées comme persistantes car elles se propagent sur le réseau, où chaque noeud conserve et contrôle ses enregistrements. Tant que la majorité des noeuds sont bons, la persistance est maintenue (VIRIYASITAVAT & HOONSOPON, 2019a).
- **Immuabilité** : chaque bloc de la blockchain est associé à un hash unique, qui dépend des informations sur ses transactions et d'autres paramètres stockés dans le bloc. Tous les blocs sont liés avec ce hachage. Si quelqu'un modifie une donnée, le hash du bloc sera modifié et la falsification des données sera détectée, ce qui rend le contenu de la blockchain immuable (HASSAN et al., 2019).
- **Anonymat et identité** : l'anonymat est la principale caractéristique des blockchains publiques. Dans la blockchain, l'identité peut être déliée de l'identité réelle d'un utilisateur car il peut créer plusieurs identités afin d'éviter l'exposition de son identité réelle. Il n'est pas nécessaire qu'une entité centrale conserve des informations privées. Par conséquent, selon les informations de la transaction, l'identité du monde réel ne peut être obtenue, ce qui préserve un certain degré de confidentialité. L'identité est généralement requise dans les systèmes qui sont exploités et régis par des entités connues dans des environnements tels que les blockchains privées et à autorisation. (VIRIYASITAVAT & HOONSOPON, 2019a ; YEOW et al., 2018)
- **Auditabilité et traçabilité** : Le timestamp des enregistrements et les informations persistantes permettent de vérifier et de retracer facilement les enregistrements précédents à travers les noeuds d'un réseau blockchain (ALI et al., 2021). Cette propriété d'horodatage rend les informations de la blockchain traçables. L'utilisateur

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

peut avoir accès à une partie des données en utilisant les timestamps et peut retracer l'historique des transactions (DAI et al., 2019). Le degré d'auditabilité dépend des types de systèmes blockchain et de leurs implémentations : Les blockchains privées sont les moins auditables car les nœuds sont administrés par une seule entité, les blockchains à autorisation viennent en second lieu, dans lesquelles certains accords, comme le cryptage des données, peuvent empêcher l'auditabilité totale des informations, et les blockchains publiques sont les plus fiables car les noeuds sont vraiment décentralisés (VIRIYASITAVAT & HOONSOPON, 2019a).

- **Sécurité** : Toutes les informations de transaction de l'utilisateur dans un réseau de blockchain restent sécurisées grâce aux différentes techniques existantes pour préserver la vie privée dans la blockchain, comme l'utilisation de chiffrement cryptographique (ALI et al., 2021).

2.3.4 Algorithmes de consensus

Un système basé sur une blockchain est aussi sûr et robuste que sa méthode de consensus sous-jacente. Il existe plusieurs méthodes bien établies par lesquelles différents nœuds d'un réseau de blockchain peuvent parvenir à un consensus sur un nouveau bloc. Dans un environnement IoT, où les appareils ont des ressources et une bande passante limitées, le choix de l'algorithme de consensus devient encore plus important. Par conséquent, nous présentons quelques méthodes de consensus existantes et discutons de la possibilité de les appliquer à un réseau IoT.

Proof of Work (PoW)

(1) Accumulation d'un lot de transactions en attente qui ont été reçues des pairs sur le réseau mais qui n'ont pas encore été incluses dans un bloc précédent et emballer ces transactions comme payload p.

(2) Recherche d'un nonce n qui, lorsqu'il est concaténé avec p, produit un hachage cryptographique qui ne dépasse pas un seuil spécifié. C'est-à-dire, $H(p - n) < t$ pour une certaine chaîne de bits t.

(3) Si un autre bloc valide est reçu avant que n ne soit trouvé, ajoutez ce bloc à la chaîne et retournez à l'étape 1.

(4) Lorsque la preuve de travail n est trouvée, diffuser le nouveau bloc, incluant n, au réseau. Retournez à l'étape 1.

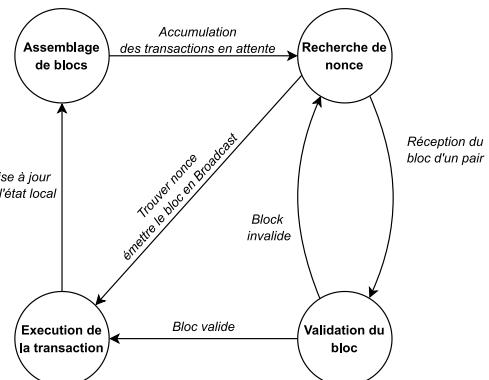


FIGURE 2.5 – Proof of Work (KOLB et al., 2020)

Le PoW est un algorithme coûteux en calcul en raison de sa technique sous-jacente de résolution d'énigmes basée sur le hachage SHA-256. Les mineurs essaient différentes valeurs de nonce pour trouver le hachage de 256 bits qui doit être inférieur (zéros à gauche) à la difficulté fixée par le réseau, qui est changée tous les 2016 blocs. Comme l'illustre 2.5, dès qu'un mineur calcule le hachage, il le transmet au réseau. Lorsque le réseau reçoit un nouveau bloc, il le vérifie. Une fois ce bloc inséré dans la blockchain, il arrête de résoudre le

CHAPITRE 2. APPRENTISSAGE FÉDÉRÉ ET BLOCKCHAIN

même bloc et passe au tour suivant (NAKAMOTO, 2008). Ce processus de recherche est le travail tandis que le nonce satisfaisant est la preuve de ce travail (KOLB et al., 2020). Avant de passer au tour suivant, la vérification du nouveau bloc par tous les mineurs est crucial pour ne pas construire de nouveaux blocs au-dessus d'un bloc non vérifié (fork) (M. KHAN et al., 2022).

Par sa conception, le PoW est constitué de retards intentionnels (latence des transactions), ce qui le rend hautement incompatible avec les réseaux IoT. La réPLICATION du registre est un autre obstacle important à l'adaptabilité de l'IoT. Enfin, le matériel spécial nécessaire pour résoudre l'éNIGME le rend résolument inapproprié pour les réseaux IoT (M. KHAN et al., 2022).

Proof of Stake (PoS)

A la place d'une compétition entre les nœuds pour résoudre le prochain bloc, un nœud est choisi par tirage au sort pour résoudre le prochain bloc. Le nœud qui extraira le prochain bloc est choisi en fonction de sa participation proportionnelle au réseau, c'est-à-dire sa richesse en termes de crypto-monnaie (SALIMITARI & CHATTERJEE, 2018). Pour qu'un nœud devienne validateur il met en jeu des crypto-monnaies comme dépôt de sécurité, par exemple en soumettant une transaction spéciale ou en invoquant un contrat intelligent (KOLB et al., 2020). Ce dépôt agit comme une garantie qu'il se comportera conformément aux règles du protocole. Le mineur peut perdre sa mise, s'il se comporte mal. Si la partie prenante a la chance de créer un nouveau bloc, elle sera récompensée (FERDOUS et al., s. d.)

PoS est une version améliorée de PoW en ce qui concerne le débit, l'efficacité énergétique et la latence, ce qui le rend favorable aux réseaux IoT. Cependant, le problème du "Nothing-atstake" est connu, car les validateurs sont sélectionnés par loterie. Si l'algorithme sélectionne un validateur avec une faible valeur d'enjeu et que celui-ci est un nœud malveillant, il ajoutera un nouveau bloc malveillant dans le registre et gagnera les frais de transaction (plus que son enjeu dans le réseau) alors qu'il n'a rien ou presque rien à risquer. La principale préoccupation de la large adaptabilité IoT est le concept monétaire, et un réseau IoT ne peut être conçu sur ce type d'enjeu (M. KHAN et al., 2022).

Byzantine Fault Tolerance

Le consensus basé sur le vote implique un vote direct pour un nouveau bloc. Il utilise l'accord byzantin (BA) pour atteindre le consensus, qui est défini comme la tolérance de panne byzantine (BFT). La BFT garantit que le système fonctionnera sans problème même s'il y a un certain nombre de nœuds malveillants (tolérance de panne). Si la majorité des nœuds sont malveillants, le système est susceptible de tomber en panne (attaque à 51%). Les systèmes basés sur BFT peuvent avoir deux types de défaillances byzantines. La première est la défaillance d'arrêt, qui peut être causée par des erreurs de réseau, comme un échec de livraison de messages ou de connexion. La deuxième défaillance est causé par le mauvais comportement d'un nœud arbitraire ou d'un nœud byzantin (nœud malveillant), par exemple, en fournissant délibérément des réponses trompeuses (M. KHAN et al., 2022).

Practical Byzantine Fault Tolerance

PBFT tente de fournir une réPLICATION pratique de la machine d'état byzantine qui peut fonctionner même lorsque des nœuds malveillants opèrent dans le système. Il fonctionne de manière séquentielle, c'est-à-dire qu'il y a un nœud leader et ensuite un nœud secondaire (backup). Le consensus commence par une demande du client (appel au consensus) au nœud leader, puis le nœud leader la diffuse aux nœuds de secours. Les deux types de nœuds traitent la demande et renvoient la décision. La demande est considérée comme complète lorsque $3f + 1$ réponses sont reçues, où f est le nombre de nœuds défectueux. Les nœuds leader et de secours peuvent être changés après chaque tour de consensus en utilisant le protocole de changement de vue (M. KHAN et al., 2022).

En raison des détails de mise en œuvre, comme le fait de ne pas utiliser de casse-tête de calcul ou d'enjeux élevés, PBFT pourrait être une bonne solution pour les applications IoT. Cependant, cet algorithme ne fonctionne bien que pour les réseaux de petite taille ou proches. En outre, il y a une surcharge de communication (diffusion de messages) qui doit être prise en compte lors de son utilisation pour les systèmes IoT. Par conséquent, le PBFT est partiellement compatible avec l'IoT.

Algorand

Algorand est basé sur PoS et la BFT (BA). Cet algorithme se concentre sur la sécurité, la décentralisation et l'évolutivité. La décentralisation est réalisée en sélectionnant aléatoirement le comité pour générer de nouveaux blocs en utilisant un protocole non interactif basé sur la clé privée et les informations publiques de chaque membre, ainsi que les enjeux associés à leurs comptes (pur PoS). La sécurité et l'évolutivité de cet algorithme sont assurées par le protocole BA, qui est utilisé pour finaliser le bloc proposé par le comité (accord byzantin). Les algorithmes BFT classiques sont vulnérables à l'attaque Sybil. Cependant, le protocole BA élimine cette menace grâce à la sélection aléatoire des comités lors de la première étape. Ce concept monétaire fait que le consensus Algorand ne convient pas pour les applications IoT (M. KHAN et al., 2022).

Proof of Elapsed Time (PoET)

Le PoET a été proposé par Intel comme mécanisme de minage alternatif en 2016. Au lieu de procéder à un minage à forte intensité de hachage, PoET simule le temps qui serait consommé par un minage PoW. C'est-à-dire que chaque nœud se retire de manière aléatoire pendant une période de temps distribuée de manière exponentielle avant d'annoncer son bloc. Pour s'assurer que le temps local est réel, PoET exige que le mécanisme de retrait soit exécuté dans un environnement d'exécution de confiance, qui est une zone de mémoire isolée assurant l'intégrité et la confidentialité du programme qui s'y exécute (XIAO et al., 2020). Les exigences de calcul réduites de PoET en font un outil idéal pour l'IoT. En outre, sa faible latence et son débit élevé le rendent favorable aux réseaux IoT. Son principal inconvénient est sa dépendance à l'égard d'Intel, qui est en contradiction avec la philosophie de base de la blockchain, entièrement décentralisée (SALIMITARI & CHATTERJEE, 2018).

Tangle

Tangle est une nouvelle technologie pour les registres distribués proposée par la crypto-monnaie Iota. Tangle ne nécessite pas de protocole de consensus compliqué, long et gourmand en ressources informatiques. Il n'utilise pas non plus de blocs pour stocker les transactions. Chaque transaction est un bloc unique en soi qui doit approuver deux transactions plus anciennes pour pouvoir être ajouté au registre. Tangle utilise un graphe acyclique dirigé (DAG) dans lequel chaque transaction est liée à deux transactions plus anciennes qu'elle a approuvées. Une fois qu'une transaction a approuvé deux transactions plus anciennes, elle est ajoutée au registre par preuve de travail (SALIMITARI & CHATTERJEE, 2018).

Tangle est construit dans l'intention de réaliser des transactions évolutives, ce qui le rend bien adapté aux réseaux IoT. Par conception, il n'y a pas de mineur et toute personne effectuant une transaction confirme que les transactions précédentes font partie du registre. Les transactions suivantes confirment ces transactions précédentes de manière parallèle, et ainsi de suite. Ainsi, chaque utilisateur est un mineur. Chaque transaction est considérée comme un nœud, et les nouvelles transactions sont liées dans le registre. Les principales caractéristiques de Tangle sont les transactions libre de droits (microtransactions), la résistance quantique et la faible latence, ce qui est souhaitable pour un réseau IoT (M. KHAN et al., 2022).

2.4 Conclusion

A travers ce chapitre, nous avons passé en revue les différents types d'apprentissage fédéré ainsi que les challenges à surmonter dans les systèmes à base de FL. De plus, nous avons vu l'intégration du FL dans les réseaux IoT et les multiples algorithmes/architectures proposés dans la littérature. Pour enfin, présenter la blockchain, qui joue un rôle très important dans la résolution des challenges vus précédemment en offrant une base solide entièrement distribuée et transparente. Afin de bien comprendre le concept de la blockchain, nous l'avons défini, ainsi que ses différents types, concepts et caractéristiques, ensuite nous avons listé quelques protocoles de consensus et son application à l'IoT.

Dans le chapitre suivant, nous présentons une panoplie d'approches présentées dans la littérature, traitant de l'intégration de trois concepts vus : IoT, Apprentissage Fédéré et Blockchain.

Chapitre 3

État de l'art

3.1 Introduction

L'intégration des approches d'apprentissage fédéré (FL) pour l'internet des objets (IoT) a récemment pris de l'ampleur. En effet, au lieu d'une vision traditionnelle où les données sont acquises, stockées et entraînées au même endroit, l'apprentissage fédéré permet aux entités impliquées de participer au processus d'apprentissage en conservant les données en local. Néanmoins, l'apprentissage fédéré présente des failles. L'étape d'agrégation du modèle global se fait, la plupart du temps sur un serveur central, représentant ainsi un point de défaillance unique, aussi le FL manque toujours de mécanismes de sécurité appropriés qui garantissent l'authentification, la confidentialité et la vie privée. Pour sécuriser le FL basé sur l'IoT, l'intégration de la technologie Blockchain est sérieusement envisagée, car elle permet de résoudre de multiples challenges tels que la préservation de la confidentialité ainsi que le point de défaillance unique que présente le serveur central.

Cependant, comme nous l'avons expliqué dans le chapitre précédent, l'intégration des ces technologies comporte de nombreuses limites et plusieurs aspects sont à prendre en considération. En général les environnements IoT sont fortement hétérogènes que ce soit au niveau des capacités de calcul, de communication et d'énergie ou au niveau de la qualité et de la nature des données recueillies. L'apprentissage fédéré pour l'IoT est souvent la cible d'attaques que ce soit des fuites de données, de l'empoisonnement ou même des nœuds frauduleux / corrompus au sein du système. Afin d'éviter cela et améliorer le temps d'entraînement du modèle global ainsi que sa précision, il est nécessaire de faire une sélection minutieuse des nœuds participants et validateurs, en prenant en compte les différentes contraintes, attaques et coûts induis.

La sélection des nœuds revêt une importance cruciale pour la réussite de l'apprentissage fédéré, étant donné qu'elle assure la garantie que les dispositifs choisis ne présentent pas de comportements malveillants et qu'ils disposent des capacités nécessaires pour contribuer de manière efficiente. Cette démarche requiert que les dispositifs sélectionnés détiennent des ressources adéquates, notamment en termes d'énergie, de mémoire, de puissance de calcul et de capacité de communication, afin de pouvoir participer pleinement à l'entraînement fédéré.

nement tout au long du processus d'apprentissage.

Dans ce chapitre, nous allons présenter une étude systémique contenant une panoplie d'articles récents qui traitent la sélection des noeuds dans notre contexte, nous expliquerons d'abord notre démarche pour le choix des approches étudiées, et nous proposerons une taxonomie de ces dernières. Nous passerons en revue chacune des approches et nous réaliserons une comparaison entre elles. Pour clôturer le chapitre, nous analyserons et discuterons des questions ouvertes pour les recherches futures.

3.2 Méthodologie de recherche

Le travail présenté dans cet état de l'art est le fruit d'une démarche méthodologique de recherche documentaire. Nous avons commencé par une une sélection d'une problématique bien ciblée et qui traite un problème précis dans le contexte étudié. Pour effectuer cela, nous avons défini la question de recherche ainsi : "**Quelles sont les méthodes de sélection des noeuds dans l'apprentissage fédéré pour les dispositifs IoT ?**"

Avec cette question en tête, nous avons entamer notre analyse documentaire en utilisant des bases de données en ligne telles que Google Scholar, IEEE Xplore, ACM Digital Library, SNDL, etc. Mais avant d'entamer la recherche nous avons défini les mots clés du sujet : Apprentissage Fédéré, Blockchain, IoT, sélection des noeuds ...etc. Ensuite, nous avons formulé les requêtes en combinant les mots clés les plus pertinents et en utilisant les connecteurs tel que ET et OU. En plus des requêtes traitant directement la problématique, nous avons analyser aussi des surveys et différentes taxonomies, afin de couvrir toutes les méthodes existantes dans la littérature. Après cela, nous avons eu plus de connaissance sur les différentes approches existant, ce qui nous a permis de s'orienter plus vers chacune des méthodes, par exemple pour trouver des travaux qui utilisent une approche bien précises, nous utilisons les mots clés reliés à cette approche dans notre requête.

Etant donné que ce sujet est assez récent, la plupart des travaux existants le sont aussi, donc les évaluer uniquement sur la base du nombre de citations serait sous-estimer certains travaux. Notre principal critère de sélection a donc été la pertinence et la renommée des revues dans lesquelles les articles ont été publiés, sur la base de leurs descriptions et s'ils traitent de la même problématique.

Après cette première sélection d'articles, nous les avons évalué en fonction de leur pertinence, leur crédibilité et leur qualité, tout en prenant des notes et en documentant nos conclusions. Ensuite, nous avons effectué une seconde sélection d'articles, qui contient que les plus pertinents et qui couvrent la totalité des approches.

La dernière étape consiste à analyser et à synthétiser les résultats trouvés, ce qui a produit le travail qui sera présenté dans les parties suivantes.

3.3 Travaux relatifs à la sélection des nœuds

Après une étude approfondie de ce qui existe dans la littérature, nous avons réussi à distinguer et différencier toutes les approches existantes pour la sélection des noeuds dans un environnement FL IoT. En conséquence, nous avons constaté que certaines reposent sur la disponibilité des ressources, d'autres prennent en considération les comportements antérieurs et la réputation. De plus, de multiples méthodes et moyens ont été utilisés pour établir la sélection, certains utilisent l'apprentissage automatique et d'autres des méthodes et algorithmes plus simples comme des approches probabilistes ou des algorithmes heuristiques. Par conséquent, nous proposons notre propre classification qui répond à la problématique dans notre contexte.

Nous avons donc divisé les approches existantes en deux grandes familles : les méthodes intelligentes avec apprentissage automatique qui sont principalement composées de méthodes d'apprentissage par renforcement, et les autres méthodes dans la famille algorithmique. Vu que les méthodes algorithmiques utilisent différentes approches avec différents objectifs de sélection, nous nous sommes concentrés sur les techniques de sélection des nœuds pour catégoriser les approches, ainsi il y a celles qui prennent en compte la réputation ou le calcul d'un certain score qui représente la fiabilité des nœuds et il y a les méthodes qui évaluent ce que les nœuds peuvent offrir comme ressources ou données. Pour chacune de ces approches, nous avons sélectionné quelques articles pour avoir une vue d'ensemble sur ce qui est proposé dans la littérature. Cette classification est présentée dans la figure 3.1 suivante :

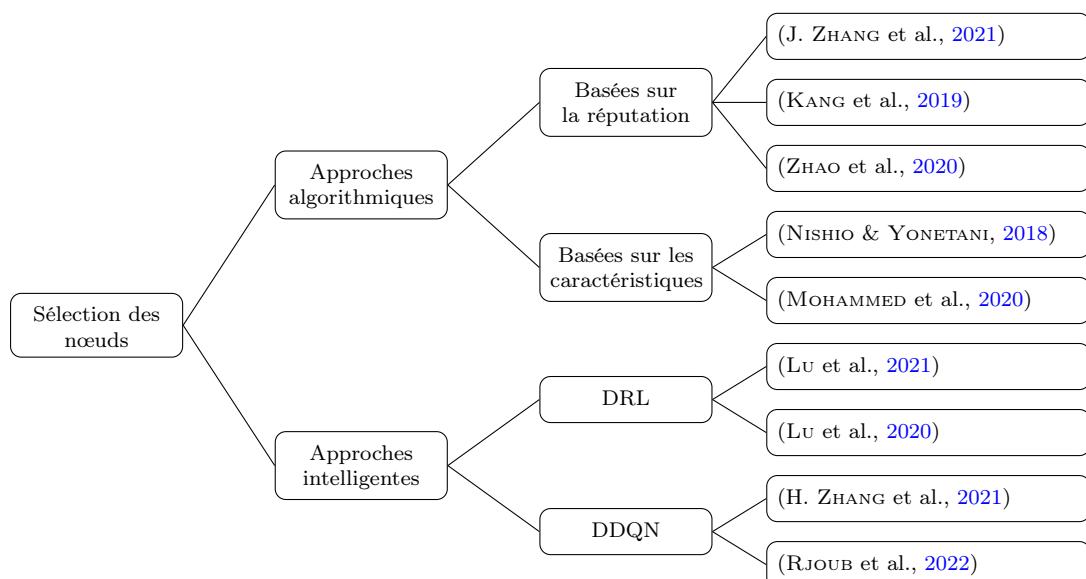


FIGURE 3.1 – Taxonomie des travaux relatifs à la sélection des nœuds.

Cependant, il est important de noter que ces approches sélectionnées ne couvrent pas la totalité des méthodes existantes pour la sélection des noeuds dans un apprentissage fédéré, mais c'est les seules méthodes qui cherchent à sélectionner des nœuds fiables et qui permettent d'améliorer la qualité de l'apprentissage. Nous nous sommes intéressés

plus par les contributions qui répondent à la même problématique que nous et avec des objectifs similaires. Parmi les autres méthodes existantes et qui ne sont pas considérées, nous citons :

- La sélection aléatoire qui n'est pas vraiment considérée comme une contribution telle quelle. Elle est surtout utilisée comme base de référence pour les expériences visant à mesurer l'amélioration apportée par d'autres méthodes.
- La sélection basée sur le regroupement (group/clustering based selection), cette méthode divise les participants dans des groupes selon différents critères, cela peut être pour des raisons de localisation (ABAD et al., 2020), pour optimiser le délai de communication (CHAI et al., 2020), pour pouvoir générer des modèles personnalisés (GHOSH et al., 2022), ou pour résoudre les problèmes imposés par les données non-IID (Z. LI et al., 2022).

3.3.1 Approches Algorithmiques

Les approches algorithmiques sont plutôt diverses et variées, dans le sens où chaque contribution sélectionnée apporte une démarche différente ou cherche à répondre à un objectif différent. Ces approches peuvent être divisées en deux grandes familles : celles basées sur le calcul de réputation ou de score, et celles qui se basent sur les caractéristiques seulement et utilisent les algorithmes heuristiques.

Approches basées sur la réputation ou le score

Pour inciter les nœuds à participer et à assurer la rentabilité des deux parties, les participants et celui qui initie le FL, les auteurs dans (J. ZHANG et al., 2021) ont proposé RRAFL, un mécanisme d'incitation basé sur la réputation et les enchères inversées. La réputation permet d'évaluer les participants et montre à quel point ils peuvent être fiables. De plus, ils utilisent la blockchain pour assurer une gestion d'une manière transparente et garantir l'immuabilité des valeurs de réputation. L'encherère inversée est une enchère où il y a un acheteur et plusieurs vendeurs, les candidats proposent des offres en fonction de leurs ressources et la qualité de leurs modèles, et le demandeur sélectionne les prix les plus bas et les meilleures scores de réputation, sous condition de ne pas dépasser son budget.

RRAFL est désigné pour un apprentissage fédéré horizontal asynchrone, et dans un contexte où plusieurs tâches de FL ont été déjà complétées et que n'importe quel membre de la communauté qui possède des données peut lancer une tâche d'apprentissage ou bien participer aux tâches des autres. Le processus de sélection des nœuds commence après le lancement de la tâche d'apprentissage par l'envoi des informations (budget, catégorie des données, ressources de calcul, etc.) à toute la communauté. Ensuite, les individus intéressés, appelés candidats, formulent leurs offres en se basant sur ce qu'ils peuvent offrir comme qualité/quantité de données et ressources. Par la suite, le demandeur récupère les réputations de tous les candidats dans toutes les tâches FL existantes depuis la blockchain, nommée blockchain d'interaction, pour pouvoir ensuite calculer la réputation compréhensive. Une fois la réputation de tous les candidats calculée, il sélectionne les candidats avec les plus bas prix et les réputations les plus élevées.

CHAPITRE 3. ÉTAT DE L'ART

La réputation compréhensive est une somme pondérée de la réputation directe et la réputation indirecte. La réputation directe est une moyenne pondérée exponentielle (l'utilisation de l'exponentielle donne un poids plus important aux valeurs récentes) de la réputation du candidat pour toutes les tâches du demandeur en question, et la réputation indirecte représente l'évaluation de la réputation par les recommandateurs effectifs, qui sont les autres demandeurs de tâche FL dont le candidat à participer. Ils sont appelés effectifs si leur évaluation de la réputation est acceptée, cela est fait en calculant la réputation des recommandateurs et le degré de similitude entre eux et le demandeur, ainsi le recommandateur est dit fiable ou non.

L'évaluation de la réputation d'un participant par un demandeur de tâche FL est calculée à partir d'un modèle de détection de qualité qui se base sur le calcul de la contribution marginale. Cette dernière n'est que la somme des contributions du participants dans chaque tour du FL, comme illustrée dans la figure 3.2, la contribution c'est la projection du vecteur de mise à jour local par rapport à la mise à jour du modèle global final. Cette solution se focalise plus sur l'impact d'un nœud que sur son effort.

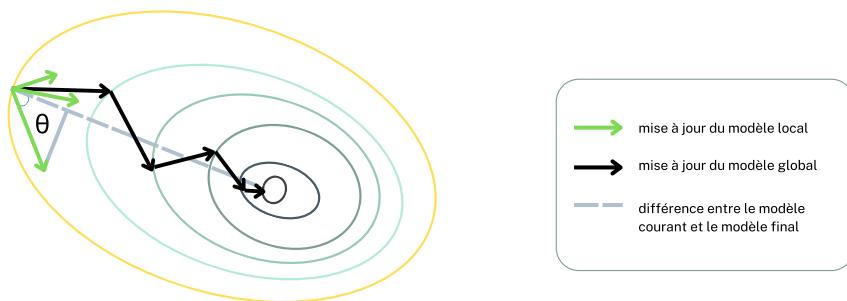


FIGURE 3.2 – Mesure de la contribution (J. ZHANG et al., 2021).

Il a été prouvé que le RRAFL est efficace du point de vue du calcul, rationnel sur le plan individuel, honnête et réalisable en termes de budget. Cependant, même avec l'utilisation de la blockchain, le système souffre du point de défaillance unique, car celui qui lance la tâche FL s'occupe de toutes les étapes : la sélection, la vérification et l'agrégation. En considérant qu'il n'est qu'un simple nœud de la communauté, il ne pourrait pas gérer tous ces calculs à une grande échelle. En effet, le système est testé sur une communauté de 30 individus, ce qui est très peu dans un contexte d'IoT. De plus, il ne prend pas en considération l'hétérogénéité des appareils participants ou des données, ni les contraintes des réseaux IoT. Par ailleurs, l'aspect sécurité n'est pas traité, et aucun mécanisme de détection d'attaques n'est implémenté.

De manière similaire, les auteurs dans (KANG et al., 2019) ont proposé un algorithme de sélection basé sur la réputation et l'utilisation de la blockchain. Le calcul de la réputation prend en considération les mêmes facteurs, tels que la réputation indirecte, la similitude avec les recommandateurs, la disponibilité des ressources, la qualité des données c-à-d la précision du modèle, la fraîcheur de la réputation (les plus récentes ont plus d'impact que les plus anciennes) et l'impact du type de l'interaction (les interactions malveillantes ont plus d'impact sur la réputation que les interactions positives).

CHAPITRE 3. ÉTAT DE L'ART

En revanche la différence entre ces deux contributions est la méthode de calcul, dans (KANG et al., 2019), ils utilisent une méthode probabiliste pour le calcul de la réputation composite, c'est le modèle logique subjectif à poids multiples (Y. LIU et al., 2011). Pour générer les opinions de réputation (l'évaluation du participant par l'initiateur de la tâche FL) ils ont proposé un mécanisme d'évaluation de qualité avec deux schémas de détection d'attaques : RONI (SHAYAN et al., 2018) qui est un schéma typique de détection d'attaque par empoisonnement, il calcule la contribution de chaque mise à jour du modèle local en comparant le modèle global avec et sans la mise à jour local pour chaque participant, et le deuxième schéma, FoolsGold (FUNG et al., 2018), il permet de détecter les mises à jour répétées en comparant les mises à jour du descente de gradient d'apparence similaire dans le cas des données non-IID. Contrairement à RRAFL (J. ZHANG et al., 2021) qui calcule la contribution une fois l'apprentissage terminé, l'évaluation de qualité dans (KANG et al., 2019) s'effectue à chaque mise à jour, ce qui impacte la complexité du système et sa scalabilité.

Pour le mécanisme d'incitation, ils ont proposé la théorie des contrats qui divise les participants sur plusieurs types. Selon ce qu'ils peuvent offrir comme ressources et qualité chaque type a une récompense adéquate. Ensuite les participants choisissent le contrat qui leur correspond et qu'ils peuvent satisfaire. Cette méthode a été prouvée équitable et offre la solution la plus rentable pour toutes les parties prenantes. Elle a été comparée au mécanisme du jeu de Stackelberg (HOU et al., 2017), qui est utilisé dans d'autres solutions de l'état de l'art, et les résultats montrent que la théorie des contrat est plus performante.

Contrairement aux deux approches déjà vues, les auteurs dans (ZHAO et al., 2020), se sont focalisés plus sur l'aspect sécurité et préservation de la confidentialité. Pour résoudre la vulnérabilité du point de défaillance unique, ils proposent un système entièrement distribué à l'aide de la blockchain. ils proposent un contexte d'appareils électroménagers, où les différentes parties prenantes sont les fabricants qui souhaites améliorer leurs produits, et les clients qui possèdent ces produits et s'occupent de l'entraînement et de l'agrégation globale.

Le processus qu'ils ont proposé peut être résumé ainsi : au début, Les fabricants lancent le crowdsourcing pour le FL, où le modèle initial avec des caractéristiques aléatoires est stocké sur la blockchain et téléchargé par les clients participants. Ces derniers collectent les données des objets connectés via leurs mobiles, et les utilisent pour entraîner un modèle local avec un CNN. Le processus d'entraînement se déroule en deux phases : entraînement mobile puis entraînement sur MEC, en introduisant une perturbation epsilon-DP pour préserver la confidentialité des données (DWORK et al., 2006). Une fois le modèle prêt, il est chargé sur la blockchain (hashé) avec une signature du client, et si celle-ci est valide, la transaction est effectuée. Le processus de consensus utilisé est Algorand (GILAD et al., 2017 ; W. WANG et al., 2019). C'est un algorithme qui est basé sur Proof of Stake, il utilise "Verifiable Random Functions" pour sélectionner un sous-ensemble de candidats qui forment un comité et un leader temporaire, et il s'appuie sur l'algorithme BFT pour valider les transactions. Après la sélection d'un leader et d'un comité, le modèle global est mis à jour à partir des modèles locaux vérifiés et téléchargés sur la blockchain. Les clients récupèrent la nouvelle version du modèle global, et le processus

CHAPITRE 3. ÉTAT DE L'ART

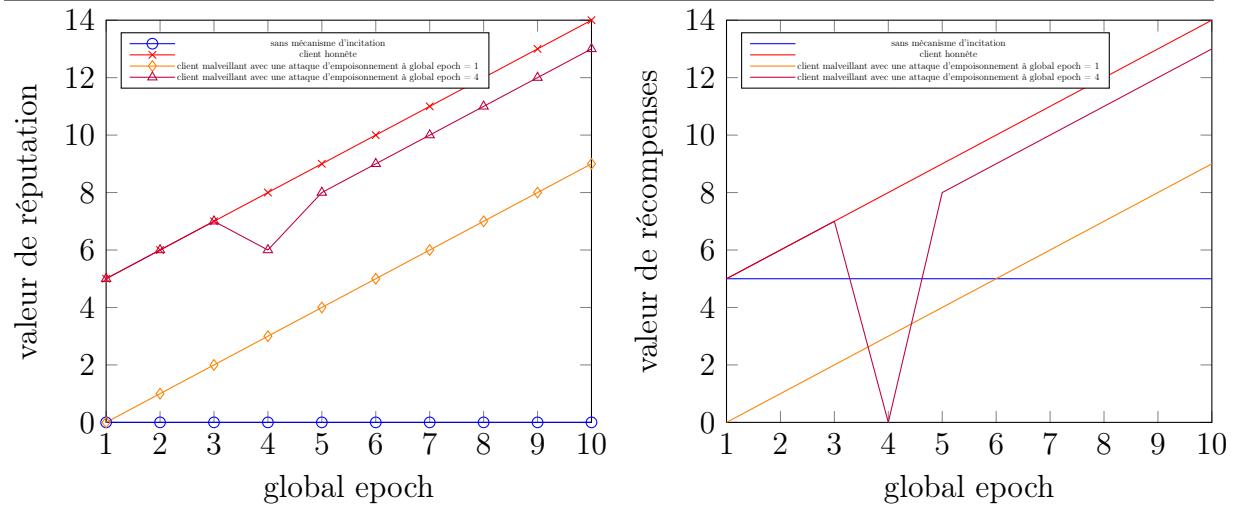


FIGURE 3.3 – Comparaison des valeurs de réputation et de récompenses

recommence.

Ils ont conçu un mécanisme d'incitation et de sélection des participants en combinant l'algorithme MULTI-KRUM proposé dans (BLANCHARD et al., 2017; SHAYAN et al., 2018) et une approches basée sur le calcul de la réputation proposée dans (Y. ZHANG & VAN DER SCHAAAR, 2012), qui est relativement simple comparée aux autres méthode de calcul de la réputation. A l'aide de MULTI-KRUM les mineurs sélectionnés vérifient la qualité des mise à jours et ils éliminent les mises à jour malveillantes : ils calculent le score de chaque client à partir de la somme des distances euclidiennes entre chaque mise à jour du client et les mises à jours $R - f - 2$ les plus proches, où R représente le nombre des mises à jours et f le nombre des clients byzantins, ensuite ils sélectionnent les $R - f$ clients avec les scores les plus bas, et le reste sera rejeté. La valeur de la récompense est proportionnelle à la réputation, cette dernière n'est qu'un score qui est incrémenté/diminué par 1 si le client est sélectionné ou non, la figure 3.3 est un exemple illustrant la valeur de réputation et de récompense des différents types de client. Les clients peuvent réclamer leurs récompenses par la suite grâce à la traçabilité des transactions sur la blockchain. Ces récompenses peuvent être des services de maintenance ou une modernisation des produits ou autres services offerts par les fabricants.

Approches basées sur les caractéristiques

Les approches basées sur les caractéristiques ne considèrent pas le comportement des noeuds ou leurs honnêtetés, ils s'intéressent seulement à ce que les noeuds peuvent offrir comme ressources de calcul et de communication et/ou la qualité des données. Généralement, les solutions proposées dans la littérature sont des algorithmes heuristiques qui cherchent à optimiser un problème formulé selon les différents objectifs.

Dans (NISHIO & YONETANI, 2018), les auteurs cherchent à optimiser le temps du processus d'apprentissage fédéré (c-à-d accélérer le temps de convergence du modèle glo-

CHAPITRE 3. ÉTAT DE L'ART

bal) dans un contexte de clients hétérogènes et à contraintes. Ils proposent le protocole FedCS, un algorithme de sélection et de planification des nœuds, qui permet à l'agrégateur de sélectionner le maximum de nœuds avec les capacités de calcul et de communication suffisantes pour pouvoir entraîner leurs modèles dans un temps limité. Ils supposent que si suffisamment de participants sont sélectionnés, les mises à jour de mauvaise qualité ou les attaques d'empoisonnement n'auront pas un impact considérable. La figure 3.4 résume les différentes étapes du protocole FedCS.

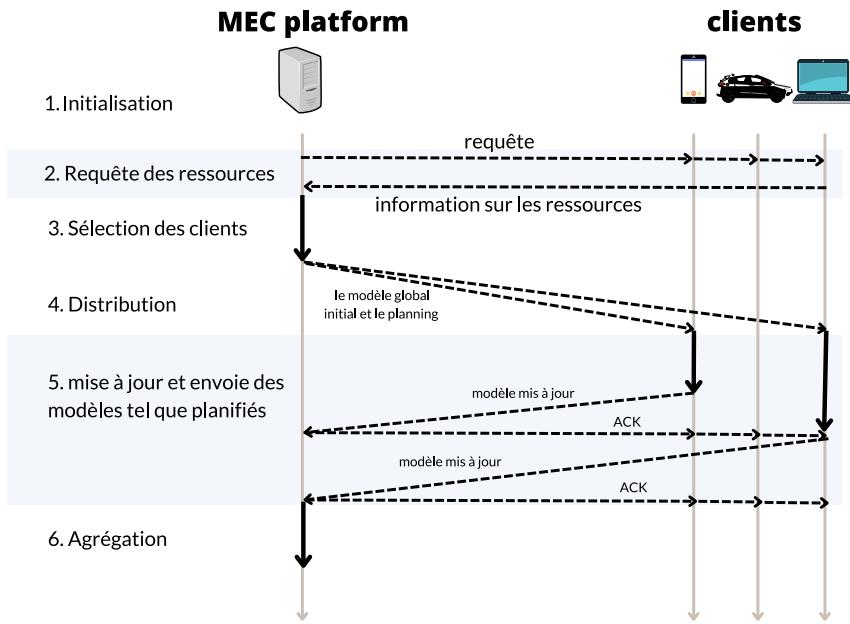


FIGURE 3.4 – Protocol FedCS (NISHIO & YONETANI, 2018).

En ce qui concerne la sélection des participants, ils ont formulé un problème d'optimisation, qui cherche à maximiser le nombre de nœuds sélectionnés sous condition de ne pas dépasser le temps fixé pour chaque tour du FL. Et pour résoudre ce problème, ils proposent un algorithme heuristique basé sur l'algorithme glouton pour un problème de maximisation avec une contrainte de type sac à dos (Knapsack) (SVIRIDENKO, 2004). En résumé, cet algorithme sélectionne les participants qui consomment le moins de temps d'apprentissage et de communication jusqu'à ce que le temps écoulé atteigne le temps limite du tour T_{round} . En revanche, le challenge de cette solution est de trouver la valeur du T_{round} la plus optimale, car plus sa valeur est grande, plus de participants sont impliqués et donc plus de données entraînées, par contre moins d'étapes d'agrégation sont effectuées, donc il faut trouver un compromis entre le nombre de participants et le nombre de tours c-à-d le nombre d'agrégations.

Parmi les inconvénients de ce protocole, nous citons le problème de point de défaillance unique avec l'utilisation d'un serveur central, l'aspect sécurité qui n'a pas été traité entièrement, et la rigidité du système qui ne s'adapte pas aux situations du monde réel : ils supposent que tous les dispositifs ont accès à une connexion sans-fil stable et ils éliminent carrément la possibilité qu'un participant sélectionné abandonne le processus, comme la figure 3.4 montre, la planification de l'envoi des mises à jour est faite dans un

CHAPITRE 3. ÉTAT DE L'ART

ordre précis. Si un participant abandonne l'entraînement, le serveur central ne pourra pas recevoir son modèle, ce qui bloque ainsi tout le processus.

Contrairement à FedCS qui s'intéresse à la quantité des participants, les auteurs dans (MOHAMMED et al., 2020) s'intéressent plus à la qualité des nœuds sélectionnés. Ils proposent une solution heuristique en temps réel pour la sélection optimale des clients en fonction de la précision de leurs modèles, inspirée du problème du secrétaire. Ils ont un budget de R clients qui doivent être sélectionnés à partir d'un ensemble de N clients qui sont accessibles temporairement, c'est-à-dire qu'ils supposent que tous les clients ne sont pas disponibles en même temps, et une fois qu'un client devient en ligne, une décision doit être prise pour savoir s'il doit être sélectionné ou non sans connaissance préalable des clients à venir.

Le problème du secrétaire est un problème classique de la théorie de la décision dans lequel un décideur doit choisir la meilleure option parmi un ensemble fini d'alternatives sur la base d'informations partielles. Il s'agit d'interviewer n candidats pour un poste de secrétaire, un à la fois, et de prendre une décision après chaque interview. La stratégie optimale consiste à rejeter la première fraction $1/e$ des candidats (THOMAS, 1989), puis à choisir le premier candidat qui est meilleur que tous les candidats précédents.

Cependant, l'algorithme qu'ils ont proposé est un peu différent de ce qui existe dans la littérature dans la mesure où ils trouvent la position d'arrêt optimal α qui permet de maximiser la probabilité de sélectionner les R meilleurs candidats, selon l'équation suivante :

$$\alpha = N \exp \left(- \left(\frac{r_2!}{(r_1 - 1)!} \right)^{\frac{1}{r_2 - r_1 + 1}} \right) \quad (3.1)$$

Où r_1 représente le minimum de R à sélectionner et r_2 représente le maximum.

Après le calcul de α , le serveur envoie les paramètres du modèle global aux clients dès qu'ils deviennent disponibles. Les α premiers clients seront rejettés automatiquement et le serveur teste leurs modèles avec le data-set de test et il récupère la meilleure valeur de précision du modèle, qui sera utilisé par la suite comme seuil pour sélectionner les R meilleurs candidats parmi les $N - \alpha$ candidats restants. La figure 3.5 est un exemple illustrant la sélection de 2 candidats parmi 10 avec cet algorithme.

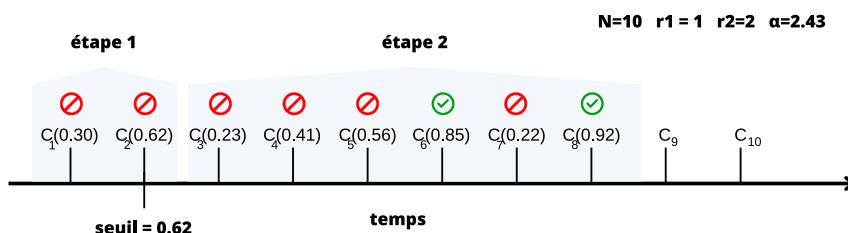


FIGURE 3.5 – Exemple d'exécution de l'algorithme heuristique du problème de la secrétaire (MOHAMMED et al., 2020).

Leur algorithme a été testé dans une application d'alarme de détection d'intrusions dans un contexte d'appareil IoT, et cela en entraînant un modèle de classification des

CHAPITRE 3. ÉTAT DE L'ART

types d'appareils IoT. les résultats montrent que l'algorithme de sélection est performant dans avec des dizaines de clients, de même qu'avec des centaines ou milliers de clients. Cependant, le problème de point de défaillance unique et l'aspect sécurité n'ont pas été traités.

3.3.2 Approches intelligentes

Dans la littérature, il existe plusieurs méthodes avancées pour sélectionner les nœuds de manière intelligente. Les méthodes qui se basent sur l'apprentissage par renforcement se distinguent particulièrement des autres. Et ce, en raison du fait que l'apprentissage par renforcement ne nécessite pas de données de pré-entraînement pour fonctionner. Il peut être déployé et utilisé sans avoir besoin d'une énorme quantité de données pour l'entraîner. Ce qui est le cas dans un processus de sélection dans un environnement FL IoT, nous n'avons aucune connaissance préalable des participants possibles, et donc l'utilisation de toute autre méthode d'apprentissage automatique n'est pas réalisable. Cependant, il est important de savoir que même si l'apprentissage par renforcement ne nécessite pas de pré-entraînement, il lui faut un certain temps de travail pour être performant et apprendre à prendre les bonnes décisions. Après une étude prolongée sur ce qui existe dans la littérature, nous avons pu identifier deux types d'apprentissage par renforcement qui sont les plus utilisés : le Deep Reinforcement Learning DRL, et le Double Deep Q-Network DDQN.

Approches basées sur l'apprentissage par renforcement profond

Pour améliorer la sécurité et la confidentialité dans les réseaux 5G et au-delà, les auteurs dans (LU et al., 2021) proposent une architecture d'apprentissage distribuée sécurisée en intégrant la blockchain dans l'apprentissage fédéré en tant que serveur de paramètres intermédiaires. Cette architecture est générique et peut être utilisée dans différents scénarios et applications, tels que : les transports intelligents, les réseaux mobiles, les scénarios IoT, le partage des ressources, Distributed D2D (Device to Device) Caching, ou bien le Edge Computation Offloading.

L'architecture proposée dans (LU et al., 2021) est représentée dans la figure 3.6, elle a trois acteurs principaux : les nœuds Edge, ce sont les clients qui entraînent les modèles locaux, ils peuvent être des IoV, IoT, appareils mobiles...etc, les serveurs Edge comme les Base Stations (BSs) ou les Macro Base Stations (MBS) sont équipés par des serveurs MEC, et la blockchain des paramètres sera utilisée pour garder trace des mises à jour des paramètres des différents modèles (locaux et globaux). Comme le montre la figure 3.6, au début la sélection des nœuds se fait avec un algorithme de DRL, ensuite une fois ces nœuds sélectionnés, ils récupèrent le modèle global depuis la blockchain (ce modèle est crypté et hashé avec le numéro de l'itération en cours). Après avoir fini leurs mises à jour du modèle, ils rajoutent le DP-bruit pour remédier aux attaques de fuite d'information et ils les remettent dans la blockchain pour pouvoir passer à l'étape d'agrégation. Cette dernière est faite par les serveurs edge qui sont sélectionnés avec l'algorithme de consensus Delegate Proof of Stake (DPoS). Durant l'agrégation, les vérificateurs s'occupent aussi de la vérification de la qualité des mise à jour des participants.

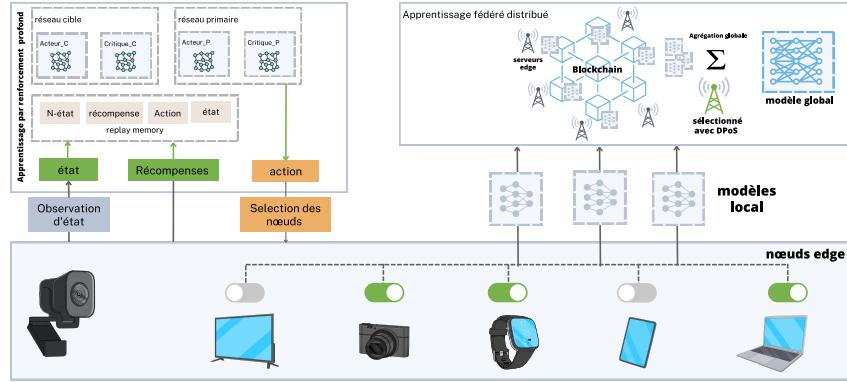


FIGURE 3.6 – Architecture d'apprentissage fédéré distribué avec sélection DRL des nœuds

Face au problème de la sélection des nœuds, ils ont formulé une équation d'optimisation combinatoire, où ils cherchent à minimiser le temps d'exécution, le coût de calcul et de communication et la perte de l'apprentissage global (c-à-d maximiser la précision du modèle global). Pour résoudre ce problème, ils proposent un algorithme d'apprentissage par renforcement profond qui est constitué de trois modules majeurs : le réseau primaire contenant deux réseaux DNN (DNN acteur et DNN critique), le réseau cible qui a la même structure, et qui est utilisé pour générer des valeurs cibles pour entraîner le DNN critique primaire, et la mémoire de répétition (replay memory).

Afin d'alléger la charge de transmission et répondre aux soucis de confidentialité et de synchronicité pour le partage des données dans un contexte d'apprentissage fédéré asynchrone pour l'internet des Véhicules (IoV), les auteurs dans (LU et al., 2020) ont proposé un système distribué et sécurisé avec l'intégration d'une architecture de blockchain hybride PermiDAG, composée d'une blockchain à permission et un graphe acyclique dirigé (DAG) local. À l'exception de cette partie, le reste de l'architecture est plutôt similaire à celle vue précédemment dans l'article (LU et al., 2021) et la figure 3.6, où les véhicules sont les clients/participants. En plus des MBS ils utilisent des RSU (Road Side Units) pour la gestion de la blockchain à permission.

Le PermiDAG, illustré avec la figure 3.7, est utilisé afin de permettre d'alléger l'agrégation globale, et cela en l'effectuant sur deux étapes, la première à travers le DAG local : chaque véhicule qui fini son entraînement local, effectue une agrégation locale avec les mises à jours vérifiées des autres participants depuis le DAG pour pouvoir améliorer sa mise à jour. Ensuite, il doit vérifier la qualité de deux mises à jours des autres pour qu'il puisse charger son modèle local sur le DAG. Après plusieurs itérations d'agrégations lo-

CHAPITRE 3. ÉTAT DE L'ART

cales et vérifications pour chaque véhicule, vient la seconde étape, une agrégation globale est effectuée par les RSUs déléguées et le modèle global généré est sauvegardé dans un nouveau bloc dans la blockchain à permission. Puis les véhicules récupèrent le nouveau modèle global et le processus réitère plusieurs fois jusqu'à l'obtention de la qualité voulue.

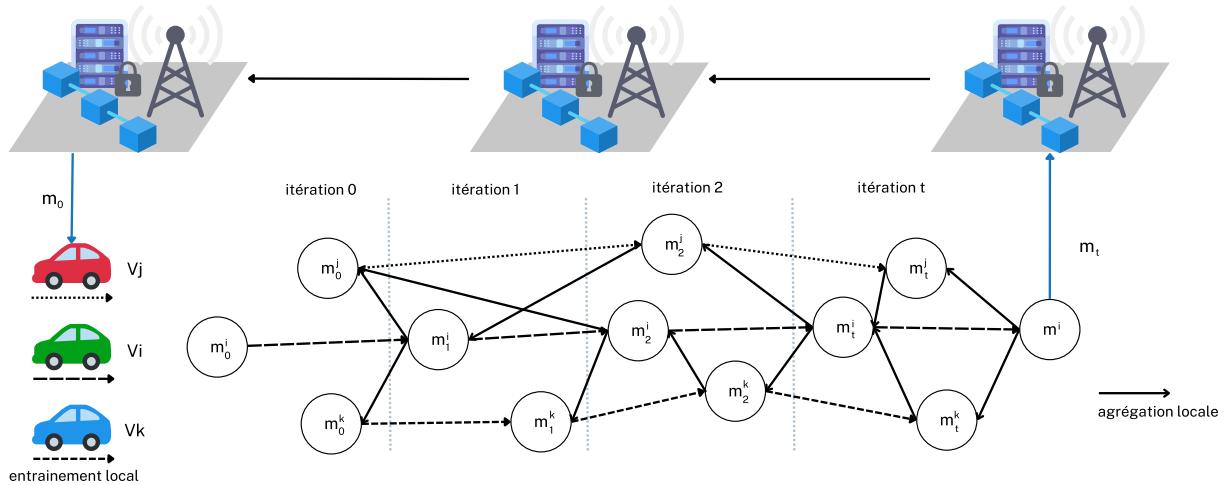


FIGURE 3.7 – Le mécanisme hybride de blockchain PermiDAG (LU et al., 2020)

Cette architecture implique plusieurs problèmes de sélection des noeuds : pour l'agrégation global, les agrégations locales au niveau du DAG, et les participants de l'entraînement FL.

- **Sélection pour l'agrégation global et la validation des transactions de la blockchain** : ils proposent un algorithme de consensus basé sur le Delegate Proof of Stake (DPoS). Cependant, au lieu de voter les délégués en fonction de leur participation (Stake), ils sont élus en fonction des véhicules sélectionnés. Les RSUs qui supportent plus de véhicules ont une probabilité plus élevée d'être élues comme délégués. Ensuite, pour chaque slot, un leader temporaire est sélectionné parmi ces délégués sur la base des performances historiques et de facteurs aléatoires.
- **Sélection pour l'agrégation locale et la validation des transactions sur le DAG** : cette sélection est basée sur le calcul de la réputation, à chaque noeuds du DAG, les auteurs associent un poids qui est proportionnel à la précision du modèle, les ressources investies pour le générer et la taille des données utilisées. Pour plus de fiabilité, ils utilisent le poids cumulatif selon la réputation des noeuds cumulés, et les noeuds avec le poids le plus lourd ont plus de chance d'être sélectionnés pour une vérification et donc pouvoir avancer plus rapidement.
- **Sélection des noeuds participants** : pour cette partie, ils utilisent un algorithme d'apprentissage par renforcement profond tels que vué déjà dans la figure 3.6. Le problème d'optimisation considéré est de minimiser le temps d'exécution et de maximiser la précision du modèle. Ils utilisent le gradient de la politique déterministe profonde (Deep Deterministic Policy Gradient DDPG) pour trouver la solution optimale pour la sélection des noeuds.

Approches basées sur l'apprentissage par renforcement double profond

Les auteurs dans (H. ZHANG et al., 2021) ont établi un nouveau système MEC pour l'apprentissage fédéré et ils ont proposé un algorithme de contrôle basé sur l'expérience qui choisit de manière adaptative les dispositifs clients qui participent à chaque tour du FL. Le système proposé est composé des clients qui sont les propriétaires des données, le serveur Edge qui est le propriétaire du modèle et une source de radio fréquence pour pouvoir charger les clients, car le type des clients considérés sont des appareils mobiles à chargement sans fil.

Le problème de sélection adaptative a été formulé comme un Markov Decision Process qui n'a pas besoin de pré-requis sur l'environnement. Pour résoudre ce problème, ils proposent un algorithme d'apprentissage par renforcement profond basé sur le double profond Q-réseaux (Double Deep Q Networks DDQN) avec l'objectif de sélectionner les participants qui consomment le moins de temps d'entraînement et de ressources telle que l'énergie. Donc, l'état des clients est représenté par la fréquence du cycle CPU, les unités d'énergie et la bande passante sans fil.

Le Deep Q-Network (DQN) traditionnel utilise un seul réseau neuronal (NN) pour prendre la décision optimale pour le serveur afin de réduire le stockage de la Q-table et d'accélérer la recherche. Cependant, le réseau neuronal unique est confronté au problème de l'estimation trop optimiste de la valeur Q, puisque le même réseau de politique sert à sélectionner et à évaluer une action pour la politique de sélection des clients. Donc pour résoudre ce problème d'estimation trop optimiste, et en se basant sur ces deux études (MNIH et al., 2015; VAN HASSELT et al., 2016), ils proposent le DDQN qui consiste en un réseau en ligne et un réseau cible. Le réseau en ligne met à jour ses poids en fonction de la relecture de l'expérience et le réseau cible réinitialise ses poids régulièrement avec l'algorithme de descente du gradient. Le DDQN sélectionne une action pour le serveur en se basant sur le NN en ligne et évalue l'action avec le NN cible qui empêche le problème de sur-optimisme de la politique.

Leur algorithme de sélection réduit la consommation d'unités d'énergie jusqu'à 50 % et le délai jusqu'à 20,70 % par rapport à l'algorithme Greedy (cet algorithme invite tous les clients à participer), tandis que le nombre de tours de communication n'augmente que de 4 %. En comparaison avec les autres méthodes de base (aléatoire, Q-learning et Greedy), leur algorithme présente de meilleures performances. Cependant, l'aspect sécurité et fiabilité des participants n'a pas été traité, de même pour la vulnérabilité du point de défaillance unique.

Le deuxième article sélectionné (RJOURB et al., 2022) pour cette approche propose aussi un algorithme de double deep Q-Networks pour une sélection des noeuds, cependant les auteurs ici s'intéressent plus à la fiabilité et la loyauté des participants. Dans un contexte d'apprentissage fédéré avec des appareils IoT, ils considèrent que la confiance entre les dispositifs IoT et le serveur edge doit être une partie intégrante du processus décisionnel de sélection. Donc ils introduisent le Trust-DDQN, un algorithme de sélection basé sur le double profond Q-apprentissage qui prend en considération le score de confiance et le niveau d'énergie des appareils IoT.

Ils définissent la notion de trust en se basant sur la consommation des ressources

CHAPITRE 3. ÉTAT DE L'ART

(CPU, mémoire et énergie). A cet effet, ils proposent un algorithme qui établie le trust en surveillant la consommation du CPU et la mémoire des appareils IoT, et ils utilisent la méthode modifiée du Z-score statistique pour détecter les activités anormales en termes de surconsommation ou de sous-consommation. Ceci est d'une grande importance pour détecter les dispositifs qui ne consacrent pas assez de ressources pour servir les tâches d'apprentissage fédéré, ainsi que ceux qui effectuent des calculs supplémentaires pour accomplir des objectifs malveillants.

En ce qui concerne la sélection des nœuds, ils formulent le problème autant que MDP, dont l'objectif est de maximiser le nombre de participants fiables et qui disposent de suffisamment d'énergie pour pouvoir entraîner le modèle et l'envoyer. Leur DDQN est plutôt similaire à celui proposé précédemment dans (H. ZHANG et al., 2021), et avec, ils utilisent le ϵ -greedy policy pour trouver la solution optimale pour la sélection des nœuds.

De plus, ils ont investigué différents algorithmes FL : FedAvg, FedProx, FedShare et FedSGD, et ils les ont comparés pour trouver celui qui est le plus approprié pour ce type de scénarios d'application. Ils ont trouvé que FedProx est celui qui donne le plus de précision et converge en moins de temps que les autres. De plus, ils ont prouvé que le trust-DDQN donne de meilleurs résultats que le DDQN classique ou l'algorithme de sélection aléatoire.

Nous arrivons à la fin de la classification des approches existantes pour la sélection des nœuds dans un contexte d'apprentissage fédéré fiable dans des environnements hétérogènes et à contraintes. Dans cette section, nous avons vu une sélection de travaux couvrant les différentes méthodes existantes, et dans la partie suivante nous allons établir une synthèse et une étude comparative de ses travaux selon différents critères.

3.4 Comparaison et évaluation des approches

Après avoir passé en revue les différentes approches sélectionnées dans la section précédente, nous allons maintenant les analyser et synthétiser les résultats, mais nous devrons d'abord identifier les critères de comparaison et les différents aspects qui doivent être examinés à travers des métriques d'évaluation. Puis nous résumerons le tout à travers un tableau de comparaison et enfin nous discuterons des résultats obtenus.

3.4.1 Métriques d'évaluation utilisées

Afin de pouvoir comparer les différentes approches, nous avons sélectionné certaines métriques et paramètres qui sont cruciales et doivent être examinées dans le contexte étudié :

Blockchain

La blockchain représente un élément essentiel dans ce contexte, car non seulement elle garantie la sécurité et la confiance entre les participants, elle permet aussi de garder trace de tout transactions et échange dans les systèmes proposées. Donc sa présence est

CHAPITRE 3. ÉTAT DE L'ART

un facteur important pour garantir une transparence dans le système. Cependant certains l'utilisent que pour le calcul de la réputation et d'autres pour décentraliser le processus d'agrégation.

Absence de point de défaillance unique (Abs PDU)

tels que mentionné dans le critère Blockchain, certains articles, même s'ils utilisent la blockchain, le problème de point de défaillance unique reste présent, et donc cet aspect du système doit être pris en considération. Nous allons les comparer selon les mesures qui ont été prises pour y remédier, cela peut être à travers différents protocoles, algorithmes de consensus, la redondance etc.

Mécanismes d'incitation

Le mécanisme d'incitation est une étape crucial dans un apprentissage fédéré. D'un coté il garantie la justesse (fairness) entre les différentes parties impliquées, et d'un autre coté il permet de recenser suffisamment de participants. Si le mécanisme proposé réussie à établir la confiance et la justesse, cela implique que moins de participants malveillants sont attirés : avec un mécanisme d'incitation efficace ils ne peuvent pas être récompensés dans le cas où ils ne travaillent pas ou bien ils effectuent des actions frauduleuses. Nous devons alors prendre en considération cet aspect du système.

Scalabilité

La scalabilité est l'une des caractéristiques impératives dans un contexte d'internet des objets, et cela est dû au nombre exorbitant d'appareils IoT impliqués. Les systèmes proposés dans ce contexte devrait s'assurer de la stabilité et la performance de leurs architecturex dans le cas d'un nombre élevé de participants. Cette métrique informe si le système supporte un nombre élevé de participants et s'il a été testé dans cette optique.

Sécurité

La sécurité est un critère important dans un apprentissage fédéré pour les réseaux IoT, car les attaquants malveillants peuvent exploiter les vulnérabilités et avoir un impact sur la fiabilité et la confidentialité du système. Cette métrique indique si la solution comprend des mécanismes pour protéger le système contre les différents types d'attaques :

- Attaques par empoisonnement : cela peut être un empoisonnement des données ou bien du modèle. Les attaques par empoisonnement consistent à injecter de fausses données dans un réseau ou une infrastructure dans le but de corrompre le modèle global. Cela peut permettre aux attaquants de voler des informations ou de modifier le processus de décision du système (C. WANG et al., 2022).
- Fuite d'information : Les mises à jour du gradient (et donc du modèle) représente une vulnérabilité car ils divulguent des informations sur les caractéristiques et les données de l'utilisateur (MELIS et al., 2019). Le principe de l'apprentissage fédéré pour la confidentialité et la protection de la vie privée est donc enfreint. L'une des

CHAPITRE 3. ÉTAT DE L'ART

méthodes pour remédier à ce problème est de rajouter du bruit (Differential Privacy) aux mises à jour. A travers ce critère nous évaluons si des mécanismes de prévention contre la fuite de données ont été utilisés ou pas.

— Détection d'anomalie :

Certains comportements anormaux peuvent être indice d'activités malveillantes ou bien d'un dysfonctionnement dans le système, et cela peut impacter les performances et la fiabilité. Ainsi la présence de mécanismes de monitoring pour la détection d'anomalie est très importante dans un apprentissage fédéré.

La considération des contraintes(**constraint-awareness**)

Les dispositifs IoT sont très contraignants en termes de ressources, d'énergie, de données et de temps de calcul. Ce critère montre donc les différentes contraintes de l'environnement qui sont prises en compte par le système, notamment, le besoin d'asynchronicité, l'hétérogénéité des dispositifs et des données, les ressources d'énergie et de communication en particulier la largeur de la bande passante.

Complexité

Vu les différentes contraintes imposées par les dispositifs IoT, et les mécanismes de sécurité qui doivent être inclus, les prendre en considération dans un système d'apprentissage fédéré implique une augmentation de la complexité du système proportionnellement à ces deux aspects. Ainsi, nous évaluons les différents articles tout en prenant en considération leur niveau de complexité. Cette métrique exprime le niveau de complexité du système que ce soit par rapport aux algorithmes ou la communication. nous avons opté pour trois niveaux de complexité : low, medium et high.

Critères de sélection des nœuds

Chacun des articles sélectionnés a pris en considération certains aspects pour leurs algorithmes et méthodes de sélection des participants¹. Il y a ceux qui s'intéressent au comportement et d'autres qui ne s'intéressent qu'aux ressources et leurs disponibilités. Les critères de sélection utilisés montrent à quel point les solutions proposées sont efficaces en terme des participants impliqués. Donc nous avons recensé tous les critères de sélection des nœuds possibles, le tableau 3.1 montre les aspects pris en compte pour chaque article.

3.4.2 Comparaison des approches

Dans ce qui suit, nous allons comparer dans des tableaux les articles sélectionnés selon les différentes métriques présentées précédemment. Le premier tableau 3.1 représente les critères² de sélection des nœuds pris en compte pour chaque article, et le second tableau 3.2 est un tableau comparatif entre tous les articles selon toutes les autres métriques.

1. le code couleurs considéré : **réputation**, **caractéristiques**, **DRL** et **DDQN**

2. ✓implique que le critère a été pris en considération.

TABLE 3.1 – critères de sélection des nœuds

Critères Articles	Réputation	Antériorité	Temps/capacité de calcul et communication	Consommation des ressources	Disponibilité des ressources	Qualité de l'apprentissage / données
art1. (J. ZHANG et al., 2021)	✓	✓				✓
art2. (KANG et al., 2019)	✓	✓	✓	✓		✓
art3. (ZHAO et al., 2020)	✓	✓	✓			
art4. (NISHIO & YONETANI, 2018)			✓		✓	
art5. (MOHAMMED et al., 2020)					✓	✓
art6. (LU et al., 2021)		✓	✓	✓	✓	✓
art7. (LU et al., 2020)		✓	✓	✓	✓	✓
art8. (H. ZHANG et al., 2021)			✓	✓	✓	
art9. (RJOUB et al., 2022)				✓	✓	

Métriques Approches	Blockchain	Abs PDU	Mécanismes d'incitation	Scalabilité	Sécurité			Contraintes prises en considération				Complexité	
					Attaques		Détection d'anomalies	Asynchronicité	Hétérogénéité		Energie	Bande passante	
					par empoisonnement	par fuite d'info			des données	des dispositifs			
art1. (J. ZHANG et al., 2021)	✓		✓		✓								Medium
art2. (KANG et al., 2019)	✓		✓		✓		✓				✓	✓	Medium
art3. (ZHAO et al., 2020)	✓	✓	✓		✓	✓				✓		✓	Medium
art4. (NISHIO & YONETANI, 2018)				✓					✓	✓		✓	Low
art5. (MOHAMMED et al., 2020)				✓						✓			Low
art6. (LU et al., 2021)	✓	✓		✓	✓	✓		✓	✓	✓		✓	High
art7. (LU et al., 2020)	✓	✓		✓	✓			✓		✓	✓		High
art8. (H. ZHANG et al., 2021)									✓	✓	✓		High
art9. (RJOURB et al., 2022)				✓			✓		✓	✓	✓		High

TABLE 3.2 – Tableau comparatif des approches étudiées

3.4.3 Analyse et discussion

A partir du tableau 3.2, nous pouvons conclure que l'utilisation de la blockchain est nécessaire pour établir un système entièrement distribué et sans point de défaillance unique. Cependant, son utilisation n'implique pas que le système soit entièrement distribué, par exemple certains travaux comme art1. (J. ZHANG et al., 2021) et art2. (KANG et al., 2019) utilisent la blockchain pour la gestion des valeurs de réputation et les autres parties du système restent centralisées comme l'agrégation et la sélection. En ce qui concerne les mécanismes d'incitation, nous remarquons que seules les approches par réputation les prennent en considération, cependant ce critère ne devrait pas être lié aux méthodes par réputation, car il doit être présent dans tout système d'apprentissage fédéré, afin de rassembler un grand nombre de participants et les récompenser pour leurs efforts de manière équitable.

Le critère de scalabilité dans notre cas signifie que l'algorithme de sélection est aussi performant avec un grand nombre de participants qu'avec un petit nombre, de même que pour la performance globale de l'entraînement : le temps global et la précision du modèle. Pour évaluer cela, nous avons examiné non seulement les résultats des tests et le nombre de dispositifs inclus, mais aussi la solution proposée et sa conception. La conclusion concernant ce critère est que, visiblement, les approches d'apprentissage par renforcement sont plus adaptatives, plus flexibles et fonctionnent beaucoup mieux avec un nombre plus élevé de participants, car elles s'améliorent avec l'expérience et les algorithmes sont plus performants au fil du temps. Quant à l'art8. (H. ZHANG et al., 2021), il a été évalué comme n'étant pas scalable, car il a traité de la scalabilité en théorie, mais sans la tester ou la prouver, et dans l'expérience, ils ont utilisé seulement 4 dispositifs IoT. Les approches heuristiques (basées sur les caractéristiques) sont également scalables, en raison de la complexité minimale de leurs algorithmes. En ce qui concerne les approches par réputation, nous constatons que tous les processus impliqués dans le calcul de réputation augmentent avec le nombre d'appareils, ce qui implique plus de ressources nécessaires. Et dans le cas d'une architecture centralisée, le serveur central peut se saturer très rapidement. De plus, la partie test de ces approches n'est exécutée qu'avec une dizaine de dispositifs IoT, alors que les autres catégories l'ont testée sur des centaines et des milliers de dispositifs.

Pour ce qui est des deux critères de sécurité et de la considération des différentes contraintes, cela relève des objectifs des différentes contributions. Certains articles ont proposé multiples schémas de détection d'attaques tel que article2. (KANG et al., 2019), art3. (ZHAO et al., 2020), art6. (LU et al., 2021) et art7. (LU et al., 2020), cependant d'autres n'ont pas pris en considération cet aspect tel que art4. (NISHIO & YONETANI, 2018) et art8. (H. ZHANG et al., 2021). A travers ce critère nous déduisons que la sécurité est un aspect très important à prendre en compte dans un tel contexte, car il permet de garantir des résultats fiables et d'établir la confiance et la confidentialité dans le système. En revanche, plus les mécanismes utilisés pour garantir la sécurité sont nombreux (par exemple les multiples étapes de vérification et test des modèles locaux tel que présenté dans art7. (LU et al., 2020)), plus le système devient complexe. Quant à la considération des contraintes, à l'exception de art1. (J. ZHANG et al., 2021) qui n'est pas destiné à

un contexte d'internet des objets, chacun du reste des articles s'est intéressé à un sous-ensemble de contraintes, mais aucun d'eux n'a proposé de solution qui tienne réellement compte de toutes les contraintes existantes dans un environnement IoT.

A propos de la complexité, nous nous intéressons aux niveaux de complexité des architectures proposées et les algorithmes de sélection plus en particulier. Les résultats obtenus indiquent que les méthodes par renforcement profond sont les plus complexes, en raison des réseaux profonds utilisés dans les algorithmes d'apprentissage automatique. Ensuite, au niveau moyen, nous trouvons les approches basées sur la réputation, cela est dû aux différentes étapes de vérifications et de calculs qui doivent être effectuées, et au niveau le plus bas de complexité, nous avons les approches heuristiques.

3.5 Conclusion

Au sein de ce chapitre nous avons vu qu'il existait différentes approches pour la sélection des dispositifs IoT pour l'apprentissage fédéré. Nous avons, tout d'abord, expliqué notre processus de recherche documentaire. Nous avons passé en revue une panoplie d'approches traitant de notre problématique, chacune son implémentation, ses forces mais surtout ses limites et faiblesses. Après avoir comparé ces dernières selon des critères bien établis, nous sommes sortis avec certaines conclusions. Les approches algorithmiques sont adéquates aux environnements à fortes contraintes, elles consomment peu de ressources de calcul cependant elles sont trop rigides, ne s'adaptent pas aux changements de comportement des noeuds et ne sont pas très scalables. Les approches intelligentes sont très adaptatives, elles peuvent être généralisées à différents cas d'utilisation et sont faites pour supporter un grand nombre d'usagers d'où leur forte scalabilité, n'empêche elles nécessitent quand même beaucoup de ressources de calcul et sont parfois très complexes.

Deuxième partie

Contribution

Chapitre 4

Conception de la solution

4.1 Introduction

L'apprentissage fédéré a été initialement introduit pour préserver la vie privée dans le domaine de l'apprentissage automatique, et il a permis d'atténuer ce problème. Cependant, de nouveaux challenges et vulnérabilités sont apparus avec cette nouvelle technique, la confidentialité des données n'est pas réellement garantie et ce à cause de l'utilisation d'un serveur centralisé qui pourrait être un grand facteur de vulnérabilité contre les attaques. En outre, l'inclusion de parties externes dans le processus d'apprentissage pourrait attirer des participants malveillants.

Ainsi, dans ce travail, nous nous penchons sur ces problématiques en proposant une architecture entièrement distribuée basée sur la blockchain et un processus de sélection minutieux des nœuds afin de garantir un apprentissage fiable et sécurisé avec des participants honnêtes. En outre, nous proposons une agrégation à plusieurs niveaux, qui non seulement assurera la disponibilité et la confiance au sein du système en éliminant le point de défaillance unique et à travers de multiples évaluations et mécanismes de détection d'attaques, mais garantira également une meilleure gestion et adaptabilité avec l'asynchronisme du système.

Dans ce chapitre, nous allons d'abord présenter une conception détaillée de l'approche proposée qui inclut le processus de travail, les modèles de communication et le modèle adversaire. Ensuite, nous introduisons le mécanisme de sélection hybride proposé. Et enfin, nous abordons le processus d'agrégation à plusieurs niveaux.

4.2 Architecture globale du système

Au sein de cette section, nous allons décrire l'environnement dans lequel notre solution sera utilisée. Nous exposerons les différentes composantes de notre système et les hypothèses le concernant. Ensuite nous détaillerons le processus de travail, puis le modèle de communication, pour enfin présenter le modèle adversaire. Le tableau 4.1 contient toutes les annotations utilisées dans la solution.

CHAPITRE 4. CONCEPTION DE LA SOLUTION

TABLE 4.1 – Sommaire des notations principales

N	Nombre de dispositifs dans le système ($n \in \{1, \dots, N\}$)
T	Nombre de tours d'entraînement (round) ($t \in \{1, \dots, T\}$)
K	Nombre de dispositifs sélectionnés pour l'entraînement durant un tour
J	Nombre de dispositifs sélectionnés pour l'agrégation et la validation durant un tour
\mathcal{B}	le budget prévu pour une tâche FL donnée
\mathcal{M}	le modèle global préliminaire à entraîner
x	Nombre de modèles à agréger dans un modèle intermédiaire
y	Nombre de niveaux d'agrégations dans l'agrégation
w_n	Paramètres du modèle local du dispositif n
w^t	Paramètres du modèle global du tour t
λ^t	Vecteur de sélection des dispositifs à un tour donné avec $ \lambda_t = N$
H_n^t	Score d'honnêteté lié au dispositif n à un tour donné t
$Ener_n^t$	Valeur de la batterie du dispositif n à un tour/round donné t
A_n^t	la disponibilité du dispositif n au tour FL t , avec $A_n^t \in [0, 1]$
Cc_n^t	Coût de communication du dispositif n à un tour donné t
Cl_n^t	Coût d'apprentissage du dispositif n à un tour donné t
f_n	Fréquence de CPU du dispositif n
r_n	Débit de transmission de communication pour un dispositif n
d_n	Taille des données du dispositif n

Notre système consiste en un réseau communautaire de dispositifs IoT hétérogènes qui travaillent en collaboration pour répondre au besoin d'un acteur externe. Ce besoin est satisfait à travers un apprentissage fédéré au sein du réseau tout en préservant la confidentialité et les données privées des parties prenantes. Au sein du réseau IoT, nous aurons une blockchain consortium qui permettra de résoudre les problèmes d'intégrité de l'IoT et de centralisation de l'apprentissage fédéré. En effet les caractéristiques inhérentes de confiance, d'autonomie et de distribution de la blockchain la rendent adaptée pour l'IoT. La blockchain permet d'établir un moyen de communication et de stockage sécurisé entre les appareils intelligents, ainsi que de les faire fonctionner de manière autonome grâce aux contrats intelligents.

Ce réseau communautaire est composé de nœuds IoT où chaque nœud n se caractérise par son identifiant unique Id_n , son débit de transmission r_n , sa fréquence de CPU f_n , l'ensemble de ses données de taille d_n , son niveau d'énergie e_n ainsi que son honnêteté h_n . Ces nœuds IoT sont catégorisés en deux principaux rôles : les nœuds blockchain qui s'occupent de la gestion et de la maintenance de la blockchain et les nœuds impliqués dans l'apprentissage fédéré. Ces derniers peuvent être soit des participants à l'entraînement soit des agrégateurs (ceux qui évaluent et agrègent les modèles) selon la sélection établie à chaque tour d'une tâche FL.

Notre système possède un acteur principal, l'initiateur de la tâche FL. Il détient son propre budget et une tâche FL bien précise. Il peut être une entreprise, le gouvernement, des établissements tels que les hôpitaux, les banques, les centres de recherche, etc. Cette

CHAPITRE 4. CONCEPTION DE LA SOLUTION

entité se doit de communiquer les informations nécessaires pour effectuer la tâche notamment le modèle global préliminaire \mathcal{M} à entraîner, le budget \mathcal{B} et l'objectif en terme de temps et/ou de qualité du modèle \mathcal{Acc} . Chaque tâche FL est effectuée sur plusieurs tour. À chaque tour une sélection est effectuée sur la communauté afin d'attribuer les tâches d'entraînement ou d'agrégation.

La figure 4.1 représente les différentes parties du systèmes ainsi que le processus de travail qui sera détaillé dans ce qui suit.

4.2.1 Processus de travail

1. Lancement de la tâche FL et initialisation des paramètres

Le processus d'apprentissage fédéré commence lorsqu'un initiateur d'une tâche FL envoie une demande à la communauté des dispositifs IoT. Cette demande contient les informations nécessaires sur la tâche, telles que le modèle global initial avec les paramètres initiaux et le budget qu'il possède pour cette tâche.

2. Mécanisme d'incitation

Afin de motiver les membres de la communauté à participer et à bien se comporter dans les tâches du FL, ils seront récompensés en fonction de leur honnêteté et de leur contribution aux tâches. Le budget offert sera réparti en fonction du nombre de tours prévus pour la tâche de FL, puis le budget du tour sera réparti entre les participants pour ce tour en fonction de leur contribution au modèle global et de leur honnêteté. Cela les motivera non seulement à participer, mais aussi à bien se comporter et à maintenir leur score d'honnêteté à un niveau élevé. Par ailleurs, la répartition du budget sur les tours permet de récompenser chaque participant juste après sa tâche, sans avoir à attendre la fin de l'ensemble de la tâche FL. Une fois le budget de la tâche connu, les nœuds de la communauté connaîtront les valeurs approximatives des récompenses et ils pourront alors décider s'ils veulent participer à la tâche ou non et s'ils veulent se candidater pour une tâche d'entraînement ou bien une tâche d'agrégation.

3. Sélection des nœuds

Une fois la liste des candidats établie avec leurs choix de tâches, l'étape suivante consiste à sélectionner les participants et les agrégateurs en fonction de leurs caractéristiques et de leur réputation. Nous proposons un seul mécanisme de sélection qui déterminera les nœuds sélectionnés et leurs rôles (entraînement ou bien évaluation et agrégation).

Il existe plusieurs méthodes de sélection des nœuds pour l'apprentissage fédéré, tels que déjà vues dans le chapitre précédent. Chacune des ses méthodes présente des caractéristiques et des avantages intéressants, afin d'en perdre aucune de ces avantages nous avons décidé de concevoir un système hybride de sélection des nœuds. Il sera basé sur deux modes de sélection, en premier lieu un algorithme, basé sur les caractéristiques des dispositifs et leur réputation, sera lancé afin d'avoir un système fonctionnel et en même temps collecter des données réelles de bonne qualité

CHAPITRE 4. CONCEPTION DE LA SOLUTION

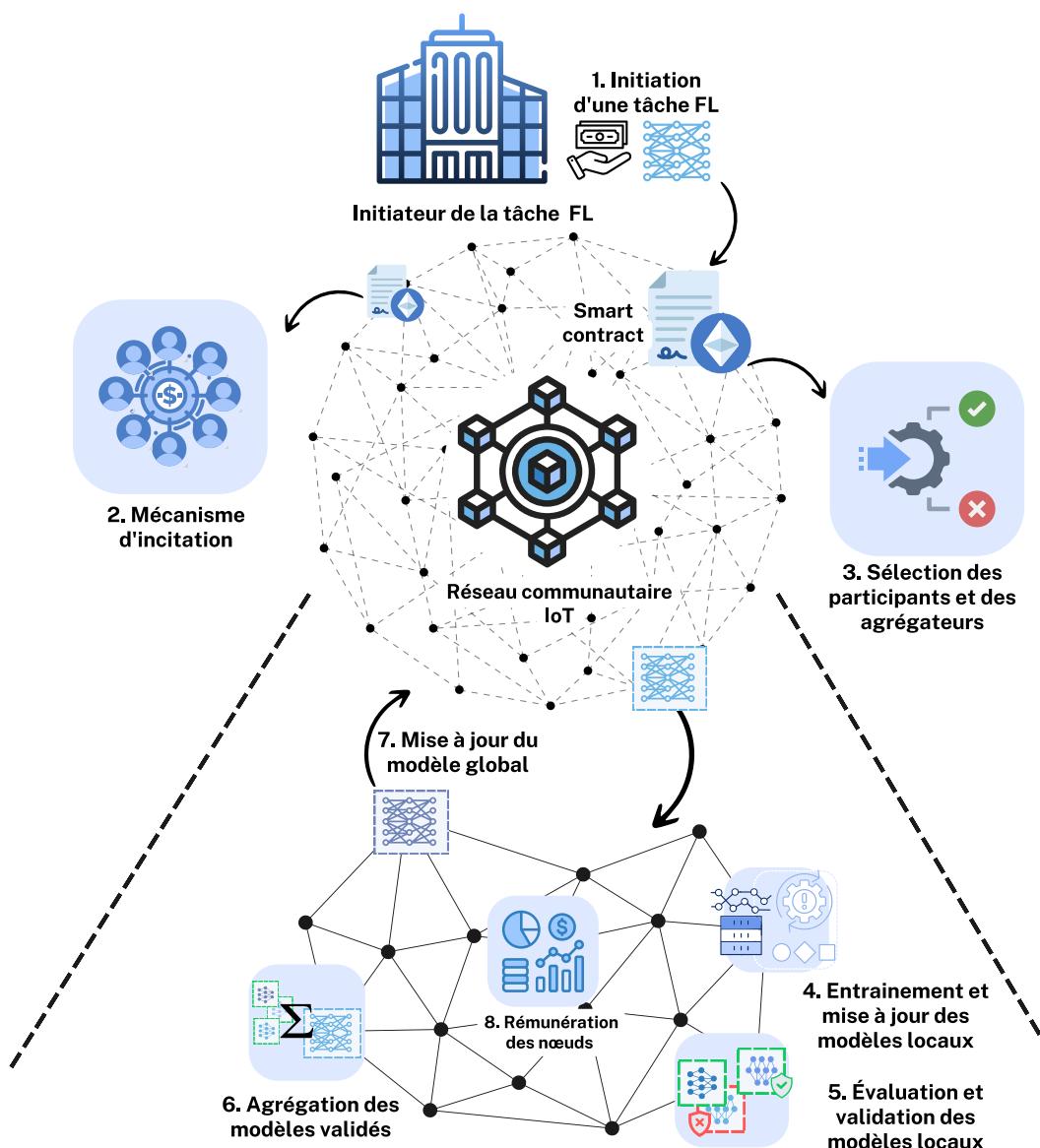


FIGURE 4.1 – Architecture globale du système.

CHAPITRE 4. CONCEPTION DE LA SOLUTION

afin d'entraîner un modèle d'apprentissage par renforcement qui sera utilisé après entraînement et convergence.

Ces méthodes de sélection sont déployés sous forme de smart contract au niveau de la blockchain et les données générées par ces smart contract seront sauvegarder au niveau de la blockchain. Toutefois, si la communauté s'agrandit et passe à l'échelle, il faudra davantage de stockage que ce qu'une blockchain peut offrir, d'où la possibilité d'envisager les solutions de stockage hors chaîne qui existent. La sélection des nœuds est au coeur de notre approche et sera détaillée plus au niveau de la section 4.4.

4. Entraînement local et mise à jour

Après la sélection des participants, ils recevront le modèle global initial ou la mise à jour pour ce tour spécifique, puis chaque appareil entraînera le modèle sur son ensemble de données local et mettra à jour les paramètres du modèle. Le modèle issu de cet entraînement est appelé modèle local. Ce dernier sera ensuite soumis à un cryptage homomorphique ainsi qu'à l'ajout d'un bruit de confidentialité différentiel (DP-noise). Ces mesures de sécurité sont appliquées pour préserver la confidentialité lors de l'enregistrement de ces modèles au niveau de la blockchain.

5. Évaluation et validation des modèles locaux

Une fois les modèles locaux insérés sur la blockchain, les agrégateurs préalablement sélectionnés commencent à les évaluer. Nous utilisons une validation multiple, où chaque modèle local est évalué au minimum deux fois : s'il est validé deux fois, il peut passer à l'étape suivante, s'il est discrédité par deux évaluateurs, il est disqualifié. Si les deux premières évaluations sont contradictoires, une troisième évaluation permettra de trancher. La valeur issue de cette évaluation multiple sera représenté par *vote* (définie dans la sous-section 4.5.2).

Pour qu'un modèle soit validé, l'évaluateur le teste sur son ensemble de données locales, puis compare sa précision à celle présentée par le propriétaire du modèle. Si une grande différence est détectée (c'est-à-dire si le participant a menti à propos de son modèle), il reçoit une évaluation négative. Si les performances du test sont nettement moins bonnes que ceux de l'apprentissage, cela peut indiquer que le modèle a été empoisonné. Et dans le cas où le modèle est évalué comme digne de confiance, c'est-à-dire sans falsification ni attaque par empoisonnement, il est validé par l'évaluateur.

6. Agrégation à plusieurs niveaux

l'agrégation se fera de manière hiérarchique, où chaque petit groupe de modèles locaux seront agrégés ensemble de manière FIFO tout en prenant en considération la disponibilité des nœuds. Cela permettra non seulement de résoudre les problèmes de ralentissement liées à l'asynchronicité et d'accélérer le processus global, mais aussi de s'adapter aux dispositifs contraignants en réduisant la quantité de modèles agrégés en une seule fois. En outre, cela décentralisera le processus d'agrégation et le

CHAPITRE 4. CONCEPTION DE LA SOLUTION

distribuera sur plusieurs nœuds du réseau, ce qui permet de résoudre le problème du point de défaillance unique. L'approche proposée est expliquée dans la section 4.5.

7. Mise à jour du modèle global

Une fois l'agrégation effectuée et le modèle global généré, cette nouvelle version est ensuite sauvegardée sur la blockchain, pour que les participants puissent la récupérer et poursuivre le processus d'apprentissage. Afin d'enregistrer cette mise à jour sur la blockchain ainsi que tous les autres mises à jour des modèles locaux, nous utilisons l'algorithme de consensus PBFT, présenté dans l'état de l'art la section 2.3.4, qui est très efficace dans les environnements byzantine.

8. Rémunération des nœuds participants

À la fin de chaque tour de FL et avant de commencer un autre tour, tous les nœuds impliqués sont rémunérés pour leur travail. Cette rémunération est proportionnelle à la contribution apportée par le nœud dans le tour FL.

4.2.2 Modèles de communication

Pour une tâche d'apprentissage fédéré, tous les travailleurs collaborent pour former un modèle global partagé et atteindre un niveau de précision global de l'apprentissage par une méthode itérative avec un certain nombre de tours de communication. Notre système intègre les principes de la mise en réseau pair-à-pair, du chiffrement et des algorithmes de consensus pour garantir une communication sécurisée et efficace. Nous avons deux types de communications :

- **Communications Device to Device** : Elles représentent les différentes communications qui peuvent avoir lieu entre les nœuds du réseau, par exemple lors d'un échange de données.
- **Transactions** : Ce sont les communications qui sont enregistrées sur la blockchain, par exemple, l'initiation de la tâche FL, la mise à jour des modèles locaux, la rémunération des nœuds.

4.2.3 Modèles adversaires

Les systèmes d'apprentissage fédéré sont vulnérables à de nombreuses attaques et présentent des failles de sécurité qui ont des conséquences majeures (BOUACIDA & MOHAPATRA, 2021). Dans ce qui suit nous allons voir les risques majeurs possibles et comment nous prévoyons de les éviter.

Les nœuds malveillants

Le premier danger auquel il faut s'attendre est la présence de participants malveillants, qui peuvent altérer la performance du modèle global à travers des attaques d'empoisonnement. Il existe deux types d'attaques d'empoisonnement, le premier étant l'empoisonnement des données, il consiste à polluer les données d'apprentissage avec des

CHAPITRE 4. CONCEPTION DE LA SOLUTION

données malveillantes, ce qui peut rendre le modèle inexact et produire des résultats erronés, le second type est l’empoisonnement du modèle, qui a pour but d’introduire des mises à jour malveillantes dans le modèle au cours du processus d’apprentissage. Pour remédier à ce genre de situation, nous concevons un mécanisme de sélection des nœuds qui prend en considération le comportement et l’honnêteté des nœuds. De plus, les mises à jour des participants passent par de multiples évaluations permettant de détecter les modèles empoisonnés. Par ailleurs, le travail des noeuds agrégateurs/évaluateurs aussi passe par des évaluations et des validations, et ces évaluations sont également incluses dans leurs scores d’honnêteté.

Fuite des données

La fuite de données dans l’apprentissage fédéré fait référence à la possibilité que des données sensibles soient exposées au cours de l’apprentissage collaboratif. Des travaux récents sur l’inversion des réseaux neuronaux profonds à partir des gradients du modèle ont soulevé des inquiétudes quant à la sécurité de l’apprentissage fédéré dans la prévention des fuites de données d’apprentissage (HATAMIZADEH et al., 2021). L’apprentissage fédéré est une méthode d’apprentissage automatique collaboratif qui respecte la confidentialité et qui nécessite des technologies de protection de la vie privée pour éviter les fuites de données lors des mises à jour des modèles locaux. Cependant, avec l’utilisation de la blockchain, les modèles sont accessibles par tous les nœuds de la communauté, et donc exposés aux attaques de fuite de données. Pour éviter cela, la présence de mécanismes de préservation de confidentialité tels que Homomorphic Encryption et Differential Privacy sont indispensables dans un système pareil (J. WANG et al., 2022) :

- **Homomorphic Encryption :** Le chiffrement homomorphe (HE) joue un rôle essentiel dans la protection de la vie privée dans l’apprentissage fédéré en permettant des calculs sécurisés sur des données chiffrées. Dans ce processus, les propriétaires de données transmettent des modèles intermédiaires, plutôt que des données brutes, à un serveur central, mais les éventuelles violations de la vie privée via les informations du modèle restent une préoccupation. Le chiffrement homomorphe résout ce problème en permettant des opérations numériques sur des données chiffrées sans décryptage. Par conséquent, le serveur central peut agréger les paramètres de modèle locaux chiffrés sans accéder à leurs valeurs réelles, assurant ainsi la confidentialité à la fois des paramètres du modèle et des données de formation locales (PARK & LIM, 2022).

De plus, l’utilisation de clés de chiffrement différentes pour les nœuds dans un système d’apprentissage fédéré renforce davantage la protection de la vie privée. Les avancées continues dans les techniques de chiffrement homomorphe, telles que xMK-CKKS (J. MA et al., 2022), offrent des performances améliorées en termes de coûts de communication et de calcul tout en préservant la précision du modèle, contribuant ainsi à la protection globale de la vie privée dans l’apprentissage fédéré.

- **Differential Privacy :** La confidentialité différentielle est une technique cruciale de préservation de la vie privée, en particulier dans le contexte de l’apprentissage fédéré. Elle offre une garantie mathématique que la présence ou l’absence de points

CHAPITRE 4. CONCEPTION DE LA SOLUTION

de données individuels dans un ensemble de données n'aura pas d'impact significatif sur les résultats des calculs effectués sur cet ensemble de données, protégeant ainsi la vie privée des individus. Pour ce faire, on ajoute un bruit calibré au cours du processus d'apprentissage, ce qui permet d'équilibrer la confidentialité et l'utilité au moyen d'un paramètre de confidentialité. Dans l'apprentissage fédéré, la confidentialité différentielle au niveau du client améliore encore la protection de la vie privée en introduisant du bruit dans les mises à jour du modèle avant l'agrégation centrale, ce qui garantit que le serveur central n'accède pas à des mises à jour précises. Les chercheurs ont développé des techniques telles que l'injection adaptative de bruit et des algorithmes d'optimisation avancés pour affiner le compromis entre la vie privée et l'utilité dans l'apprentissage fédéré avec confidentialité différentielle, ce qui en fait un outil précieux pour la préservation de la vie privée (DWORK & ROTH, 2014).

Les risques de communication

Une communication non sécurisée ou de mauvaise qualité peut engendrer des risques importants tels que les goulots d'étranglement des communications (communication bottleneck), les attaques du type man-in-the-middle ou bien des participants qui abandonnent leurs tâches d'apprentissage. L'utilisation de la blockchain permet de remédier à ce genre de problème, car elle agit comme un moyen de communication intermédiaire entre les participants et les agrégateurs. Elle garde trace de toutes les transactions au niveau du système en toute transparence, ce qui permet de détecter les cas d'abandon ou les attaques d'intrusion et, en même temps, éviter les problèmes de surcharge de communication.

4.3 Métriques de sélection des nœuds

Afin de pouvoir évaluer les nœuds et leur travail nous aurons besoin de quantifier différentes caractéristiques. Dans ce qui suit nous allons voir comment nous pouvons calculer la valeur du résultat du vote d'évaluation du modèle, le score d'honnêteté d'un nœud, ainsi que le coût global d'un round d'apprentissage.

4.3.1 Contribution

Pour le calcul de la contribution, nous utilisons la similarité en cosinus où la contribution d'un nœud participant est calculée selon l'angle de déviation du modèle local qu'il propose par rapport au modèle global. La similarité en cosinus est une mesure utilisée pour déterminer la similarité entre deux vecteurs non nuls d'un espace de produit intérieur. Elle quantifie le cosinus de l'angle entre ces vecteurs, indiquant à quel point ils sont alignés. En d'autres termes, elle mesure la similitude de direction indépendamment de la magnitude des vecteurs. Comme nous voulons calculer la contribution par rapport au tour d'entraînement, nous allons l'utiliser pour calculer la contribution par rapport au modèle global du tour t .

$$Contrib_k^t = \frac{m_k^t \cdot m^t}{\max(\|m_k^t\|_2 \cdot \|m^t\|_2, \epsilon)} \quad (4.1)$$

4.3.2 Score d'honnêteté

Ce score sera utilisé pour évaluer le travail des participants du FL et leur honnêteté. Au début de la tâche d'apprentissage fédéré, il sera initialisé à la même valeur pour tous les dispositifs $H(0)$. Pour chaque dispositif ce score se verra augmenté ou diminué selon son comportement, le type d'action qu'il a effectué (entraînement, évaluation ou agrégation) et sa contribution à chaque tour d'apprentissage, c'est-à-dire, à la fin du processus d'agrégation et l'obtention du modèle global du tour. L'algorithme 1 détaille comment le score d'honnêteté est calculé. Les facteurs α , β , γ et ϕ sont utilisées dans la formule du score d'honnêteté et leurs valeurs seraient déterminer par expérimentation. Le facteur α permet de déterminer l'impact de la contribution sur le score, β détermine l'impact de la pénalité sur le score en cas de malhonnêteté. Quant aux facteurs ϕ et γ , ils sont utilisés pour pondérer la formule de contribution des agrégateurs

Algorithme 1 : Calcul de l'honnêteté d'un participant

```

1 for chaque nœud participant  $j$  sélectionné du tour  $t$  do
2   if participant  $j$  abandonne then
3      $Contrib \leftarrow MALUS$ 
4   else
5     if  $m_j$  est validé then
6        $Contrib \leftarrow Contrib_j^t$ 
7     else
8        $Contrib \leftarrow -\beta \cdot |Acc_{eval} - Acc_j^t|$ 
9    $H_j^t \leftarrow H_j^{t-1} + \alpha \cdot Contrib$ 

10 for chaque nœud agrégateur/validateur  $k$  sélectionné du tour  $t$  do
11   if participant  $j$  abandonne then
12      $Contrib \leftarrow MALUS$ 
13   else
14      $Contrib \leftarrow \frac{(\phi \cdot évaluations vraies + agrégations vraies - \gamma \cdot (évaluations fausses + agrégations fausses))}{\sum valuations_k + agrégations_k}$ 
15    $H_k^t \leftarrow H_k^{t-1} + \alpha \cdot Contrib$ 

```

Le calcul de l'honnêteté diffère selon le type de tâche accompli par le nœud, si le nœud a effectué l'entraînement d'un modèle local, dans le cas où ce nœud n'a pas falsifié ses résultats, il sera récompensé en conséquences, dans le cas contraire il recevra un malus proportionnel à l'ampleur du sabotage intentionné. Quant aux validateurs et agrégateurs, ils seront récompensés par rapport aux agrégations et évaluations correctes qu'il auront effectuées.

Acc_{eval} est la moyenne des précisions du même modèle local issu les évaluations des agrégateurs et Acc_j^t la précision du modèle mis à jour par le nœud k . Nous avons fait en sorte d'accorder un malus qui réduit drastiquement l'honnêteté d'un nœud lorsque nous détectons une fraude. Ainsi, dans notre système, il est difficile de gagner la confiance du réseau, en revanche il est très facile de la perdre, ce qui dissuadera les nœuds de mal se conduire.

4.3.3 Coût global

Étant donné que nous travaillons avec des dispositifs hétérogènes, afin de pouvoir calculer le coût global, il faut prendre en compte le coût de téléchargement du modèle initial, le coût de l'entraînement du modèle pour chaque dispositif à chaque round et le coût de transmission des paramètres aux agrégateurs.

Tout d'abord, nous commençons par le coût d'entraînement du modèle, en sachant que chaque dispositif possède une vitesse de processeur et un dataset différents (4.2).

$$Cl_n^t = \frac{d_n \cdot \beta_n}{f_n} \quad (4.2)$$

où β_n est le nombre de cycles CPU nécessaires pour l'entraînement du modèle sur une instance du dataset.

le coût de transmission des paramètres quant à lui est lié au débit de transmission du dispositif.

$$Cc_n^t = \frac{|w_n|}{r_n} \quad (4.3)$$

avec $|w_n|$ la taille des paramètres du modèle local entraîné. Le coût global d'un round d'apprentissage, est donné par la formule qui suit

$$C^t = \max_{n \in N} (Cl_n^t + Cc_n^t) \quad (4.4)$$

NB : on considère que la vitesse de transmission en download est beaucoup plus rapide qu'en upload ce qui rend le coût de téléchargement du modèle à entraîner négligeable.

4.4 Mécanisme hybride distribué pour la sélection des nœuds

La sélection des participants est une étape cruciale durant l'apprentissage fédéré pour l'internet des objets, car elle garantit que les dispositifs IoT sélectionnés ne sont pas malveillants et qu'ils sont en mesure de contribuer efficacement. Notre objectif est de minimiser le coût global de l'entraînement et d'optimiser la convergence du modèle global tout en assurant l'honnêteté des participants. Cependant, la nature hétérogène et à forte contraintes des environnements IoT entrave l'efficacité globale de l'entraînement. Cela est dû aux ressources limitées, l'indisponibilité des appareils ou bien des mises à jour de mauvaise qualité. Par conséquent, les nœuds les plus rapides (c.à.d ils consomment moins de temps d'entraînement et de communication), les plus honnêtes, ceux avec des modèles de meilleure qualité (c.à.d une précision des modèles plus élevée) et ceux disposant de suffisamment d'énergie doivent être sélectionnés.

Afin de réaliser ses objectifs, nous concevons un mécanisme hybride entièrement distribué de sélection des nœuds, qui combine les deux approches vues dans l'état de l'art,

CHAPITRE 4. CONCEPTION DE LA SOLUTION

avec l'objectif d'exploiter les avantages de chacune de ses méthodes. La figure 4.2 explique comment ces deux approches sont intégrées dans le système. D'une part , les méthodes à base de réputation/score ne sont pas gourmandes en calcul et en données mais sont rigides devant les environnements dynamiques des IoT. Les méthodes intelligentes, malgré leur consommation en calcul et en données, sont plus adaptatives aux changements et sont adéquates aux environnements dynamiques des IoT.

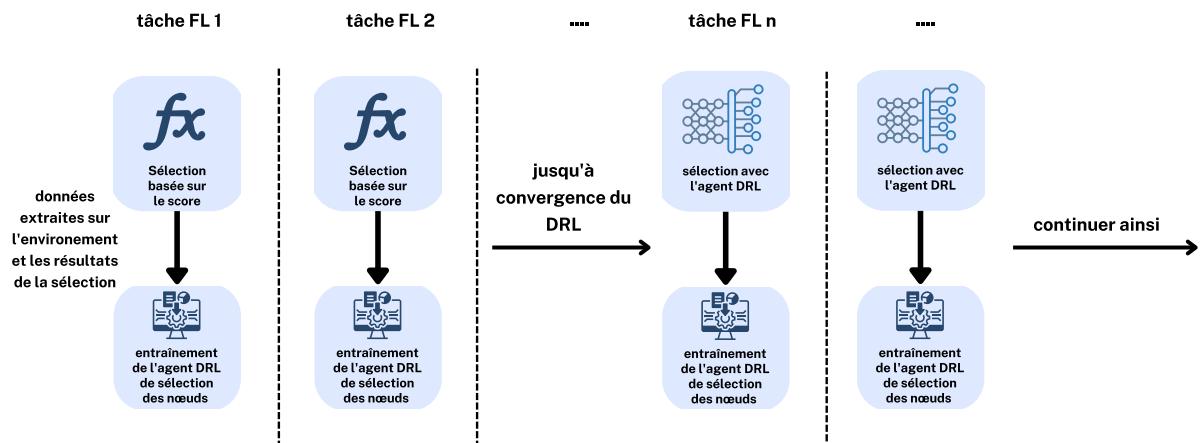


FIGURE 4.2 – Mécanisme hybride distribué pour la sélection des noeuds

La sélection proposée dans notre système passe par deux étapes. En premier lieu un algorithme, basé sur les caractéristiques des dispositifs et leur comportement et honnêteté, sera lancé afin d'avoir un système fonctionnel dès le début. Ainsi, nous pourrons collecter des données réelles de bonne qualité afin d'entraîner un modèle d'apprentissage par renforcement qui sera le deuxième mode de sélection. Dans ce qui suit, nous allons voir en premier lieu la méthode de sélection basée sur le score, ensuite nous présentons la solution DRL et enfin nous expliquerons comment nous intégrons ces deux approches ensemble dans une solution nommée hybride.

4.4.1 Méthode basée sur le score

Les méthodes exactes de sélection des nœuds sont diverses, comme nous l'avons vu dans l'état de l'art, certaines sont basées sur un calcul de score de réputation et d'autres basées sur les caractéristiques (meilleur temps d'entraînement, meilleur capacité de stockage, etc). La valeur de ce score décide de l'état de sélection du nœud : si le score dépasse un certain seuil le dispositif est sélectionné sinon son état de sélection est mis à zéro. Dans notre cas nous allons concevoir un formulaire de sélection qui prend en considération les caractéristiques du dispositifs ainsi que son honnêteté au cours de la tâche d'apprentissage fédéré.

Avant toute chose, nous posons certaines conditions pour cette sélection. Un dispositif est considéré comme non disponible pour effectuer la tâche dans le cas où il annonce lui-même son indisponibilité ou si son niveau d'énergie est inférieur à un seuil précis.

CHAPITRE 4. CONCEPTION DE LA SOLUTION

Soit $Ener_n^t$ la valeur d'énergie (batterie) du dispositif n au round t , et A_n^t sa disponibilité, donc la condition est sous la forme :

$$A_n^t = \begin{cases} 0, & \text{si } Ener_n^t \leq Ener_{min} \\ 0, & \text{si le dispositif } n \text{ annonce son indisponibilité} \\ 1, & \text{dans le cas contraire} \end{cases} \quad (4.5)$$

Avec $Ener_{min}$ l'énergie minimale nécessaire pour effectuer un round de la tâche FL en cours.

La sélection ne se fait que sur les dispositifs disponibles, c'est-à-dire ceux qui sont volontaires pour participer à la tâche d'apprentissage et qui ont assez de batterie pour effectuer le round d'entraînement. Nous favorisons les dispositifs qui détiennent le plus de ressources, les plus honnêtes et bien sur ceux qui contribueront le plus à la convergence rapide du modèle global. Soit S_n^t score du dispositif n à un tour donné t . En changeant les paramètres des littéraux de l'équation du score, nous pourrons décider de l'impact de l'honnêteté et des caractéristiques. Plus le paramètre du facteur d'honnêteté est grand par rapport à celui des caractéristiques plus la sélection se basera sur l'honnêteté et vice versa. L'algorithme 2 résume la procédure de sélection des noeuds participants au FL.

Algorithme 2 : Sélection des participants

- 1 pour chaque dispositif, sa disponibilité A est d'abord vérifiée ensuite son score S est calculé.
 - 2 **for** n in range ($1, N$) **do**
 - 3 $\lambda_n^t \leftarrow A_n^t$
 - 4 $S_n^t \leftarrow \alpha \cdot H_n^t - (1 - \alpha) \cdot C_k^t$
 - 5 Sélectionner les $K + J$ noeuds disponible avec les meilleurs scores
 - 6 Affecter les J premiers aux tâches d'agrégation et d'évaluation et le reste aux entraînements locaux
-

Ainsi nous aurons les meilleurs noeuds en termes d'honnêteté et les plus performants (moins coûteux) sélectionnés pour les tâches d'agrégation, d'évaluation et de mise à jour des modèles locaux.

4.4.2 Méthode DRL

L'apprentissage par renforcement est un type d'apprentissage automatique dans lequel un agent apprend à prendre des décisions en interagissant avec son environnement. L'agent observe l'état actuel de son environnement ensuite il exécute une action pour passer à un autre état qui lui permet de maximiser une récompense à travers le temps appelée récompense cumulative. Il apprend à prendre les bonnes décisions grâce à une politique qui associe les états aux actions, et à travers ses expériences il met à jour la politique. Le processus d'apprentissage se poursuit jusqu'à ce que la politique de l'agent converge vers une politique optimale, qui maximise la récompense cumulative attendue. Dans le cas où l'état et les actions sont multiples il est difficile pour l'agent de juger quelle action entreprendre étant donné le très grand nombre de combinaisons possibles, ce qui nous ramène à l'apprentissage par renforcement profond.

CHAPITRE 4. CONCEPTION DE LA SOLUTION

L'apprentissage par renforcement profond est un sous domaine du RL qui utilise des réseaux de neurones comme politique, permettant à l'agent d'apprendre à partir d'entrées à haute dimension telles que des images et des données brutes de capteurs.

Pour résoudre le problème de sélection des noeuds, nous devons d'abord formuler le problème en définissant les éléments d'un algorithme RL, notamment les états de l'environnement, les actions et la récompenses, ensuite nous proposons l'algorithme DRL qui permet de résoudre le problème d'optimisation formulé.

Formulation du problème

le problème de sélection des noeuds participants au FL peut être formulé par un Marcov Decision Process. un MDP est défini par $M = (E, A, P, R)$, où E représente l'espace des états du système, A est l'espace des actions que l'agent doit prendre, $P = Pr(e'|e, a)$ est la probabilité d'une transition d'un état e à un état e' en prenant l'action a et R représente la fonction de récompense.

1. Espace des états

L'état du système est représenté par l'ensemble des états des dispositifs IoT. L'état global du système E est obtenue avec un produit cartésien des états des dispositifs IoT : $E = \prod_{n \in N} E_n$, où chaque état $e_n \in E_n$ est décrit ainsi :

$$e_n^t = \{H_n^t, E_n^t, \lambda_n(t-1), Cc_n^t, Cl_n^t, A_n^t\} \quad (4.6)$$

Où H_n^t représente le score d'honnêteté du noeuds n au tour t , E_n^t est son niveau d'énergie, $\lambda_n(t-1)$ est sa valeur de sélection au tour précédent ($t-1$), Cc_n^t et Cl_n^t sont le coût de communication et le coût d'apprentissage nécessaires et A_n^t représente la sa disponibilité au tour t .

2. Espace d'actions

L'action à l'instant t est la décision de sélection d'un dispositif, qui peut être considérée comme un problème 0-1. l'action $\lambda^t \in A$ est définie par un vecteur, nommé vecteur de sélection :

$$\lambda^t = (\lambda_1^t, \lambda_2^t, \dots, \lambda_n^t) \quad (4.7)$$

où $\lambda_i^t = 1$ si le dispositif i est sélectionné pour participer au tour t de la tâche FL, sinon $\lambda_i^t = 0$.

3. Fonction de récompense

La fonction de récompense R détermine la direction d'apprentissage du DRL, car elle permet d'évaluer l'effet des actions prises.

La récompense du dispositif n a l'instant t en fonction de l'état e_n^t et l'action λ_n^t est définie ainsi :

$$R_n^t(e_n^t, \lambda_n^t) = \lambda_n^t \cdot H_n^t \cdot Acc_n^t \quad (4.8)$$

Ainsi la récompense de l'agent lorsqu'il performe une action prend en considération l'honnêteté du noeud ainsi que ces résultats (la précision de son modèle).

La récompense global du système à un instant t est défini ainsi :

$$R(e^t, \lambda^t) = \sum_{n \in N} R_n^t(e_n^t, \lambda_n^t) \quad (4.9)$$

4. Politique

Le système détermine la politique optimale $\pi^* : E \rightarrow A$ qui indique les actions à entreprendre à chaque état pour maximiser la récompense cumulée. Dans le cas d'un apprentissage par renforcement profond la politique est générée par un réseau de neurone qui permet de générer les actions à entreprendre à partir des états du système en entrée.

Algorithme DRL pour la sélection des nœuds

Lorsqu'on tente de résoudre un problème d'apprentissage par renforcement, il existe deux choix de solutions qui s'offrent à nous :

- **Méthodes value-based** : ces méthodes consistent à apprendre la value-function et leur principal objectif est de minimiser la perte entre la valeur prédite et la valeur cible.
- **Méthodes policy-based** : l'idée est de paramétriser la politique en utilisant par exemple un réseau de neurones. L'objectif de ces méthodes cherchent à maximiser les performances de la politique paramétrée,

Pour notre cas nous avons choisi d'utiliser une méthode qui combine les deux citées plus haut. Les méthodes Actor-Critic sont des méthodes de Temporal Difference (1.3.2), elles sont considérées comme une combinaison entre les méthodes value-based et policy-based. Actor-Critic dispose d'une structure de mémoire distincte représentant explicitement la politique indépendamment de la value-function.

La figure 4.3 montre les composants d'un modèle DRL de type Actor-Critic.

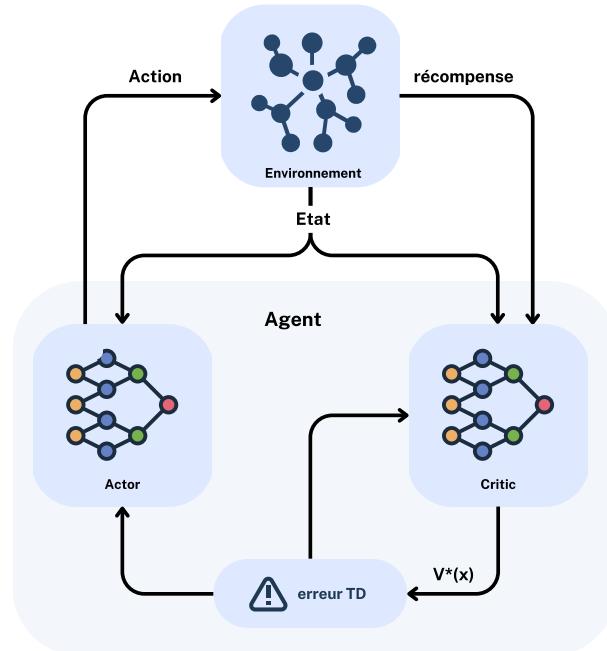


FIGURE 4.3 – Modèle d'apprentissage par renforcement profond du type Actor-Critic

CHAPITRE 4. CONCEPTION DE LA SOLUTION

L'agent est composé de deux réseaux de neurones

- L'acteur représente la politique, il est utilisé pour sélectionner les actions.
- Le critique représente la value function estimée, il critique donc les actions effectuées par l'acteur.

L'apprentissage se fait toujours dans le cadre d'une politique : le critique doit se renseigner sur la politique actuellement suivie par l'acteur et la critiquer. La critique prend la forme d'une erreur de TD. cette critique est utilisée pour mettre à jour l'agent c-à-d modifier les poids des réseaux de neurones de l'acteur et du critique.

Dans ce qui suit nous allons voir le processus de travail d'un algorithme Actor-Critic.

Pour chaque pas de temps t

- Récupérer l'état de l'environnement e^t et le transmettre en entrée à l'acteur et au critique.
- Notre politique prend l'état et produit une action λ^t .
- Le Critique prend également cette action en entrée et, à l'aide de e^t et A^t , calcule la valeur de cette action dans cet état : la Q-value.
- L'action A^t effectuée dans l'environnement produit un nouvel état S^{t+1} et une récompense R^{t+1} .
- L'acteur met à jour ses paramètres de politique en utilisant la Q-value.
- Grâce à ses paramètres mis à jour, l'acteur produit la prochaine action à entreprendre $A^t + 1$ étant donné le nouvel état $e^t + 1$.
- Le Critique met ensuite à jour ses paramètres de valeur.

4.4.3 Méthode Hybride

Comme nous l'avons pu le voir au niveau de l'état de l'art, les méthodes basées sur le score sont en quelque sorte rigide vis-à-vis des changements de l'environnement et souvent très peu scalables. les méthodes d'apprentissage par renforcement profond, quant à elles sont très adaptatives aux environnements dynamiques mais parfois prennent beaucoup de temps pour converger vers une politique optimale, surtout du fait que les données d'entraînement sont souvent issues de simulations. Dans l'optique d'avoir un système fonctionnel avec un bon rendement dès son déploiement et sans attendre qu'un modèle soit entraîné, nous avons choisi de combiner ces deux solutions. L'objectif principal de l'utilisation d'une méthode basée sur le score est donc d'avoir un système opérationnel avec de bons résultats de performance tout en collectant des données pertinentes et réelles qui serviront de données d'entraînement pour le modèle Actor-Critic.

La sélection effectuée par la méthode exacte ainsi que les états des dispositifs (entrées pour la sélection) et le résultat obtenu, qui est la fonction de perte du modèle global au tour t , seront sauvegardés au niveau de la blockchain sous forme d'un ensemble d'expériences. À la fin de chaque tâche FL, ces expériences seront appelées pour entraîner l'agent DRL en utilisant un contrat intelligent, Le processus d'entraînement de l'agent sera exécuté jusqu'à sa convergence, et à partir de ce moment-là, la sélection sera effectuée par l'agent. La figure 4.4 illustre le fonctionnement de cette méthode hybride.

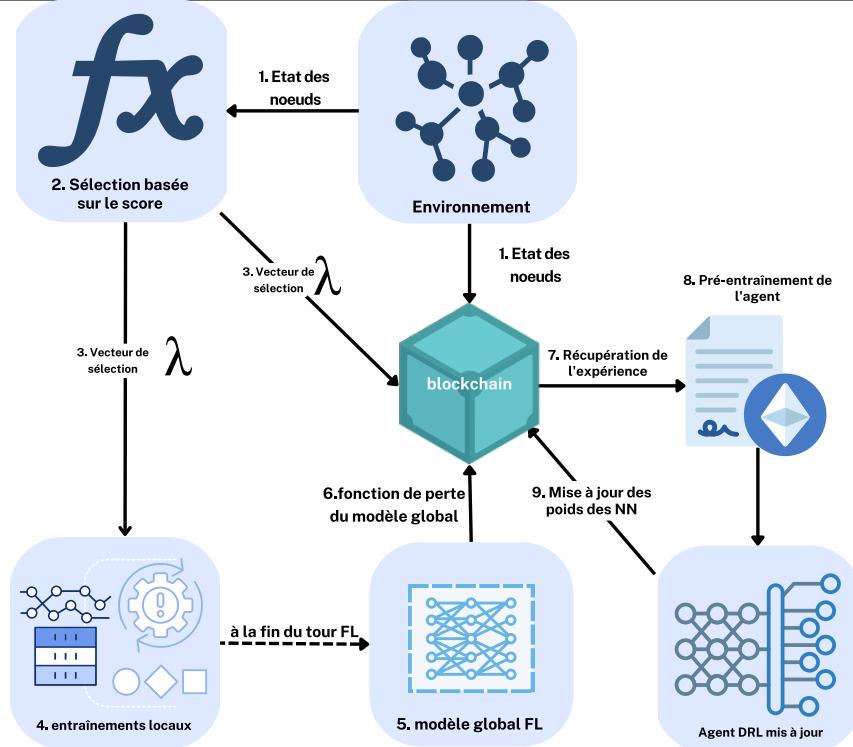


FIGURE 4.4 – Fonctionnement du mécanisme hybride distribué pour la sélection des noeuds

Comme nous l'avons vu au début de ce chapitre, après la sélection des participants vient l'entraînement local du modèle. Chaque participant entraîne le modèle fourni sur ses propres données. Dans un système d'apprentissage fédéré classique, les modèles mis à jour sont envoyés à un serveur central pour être agrégés. Afin de se débarasser du point de défaillance unique et distribuer l'agrégation nous proposons notre propre agrégation de façon incrémentale (à plusieurs niveaux). Dans ce qui suit nous allons voir en détails ce processus ainsi que tous les traitements qui viennent avec cette agrégation.

4.5 Agrégation globale à plusieurs niveaux du FL

4.5.1 Processus d'agrégation

Le processus d'agrégation commence une fois que les mises à jour des modèles locaux sont chargées sur la blockchain, et il se chevauche avec l'étape d'évaluation. En raison de l'asynchronicité et de la différence de coût/temps dans les mises à jour de certains participants, les modèles locaux ne peuvent pas être évalués et agrégés tous en même temps, sous peine de retarder considérablement le processus. Ainsi, une fois qu'un modèle local est téléchargé sur la blockchain, le premier agrégateur disponible doit l'évaluer et lui donner une validation ou bien un refus.

Avec la possibilité que des participants abandonnent le processus et ne soumettent pas

CHAPITRE 4. CONCEPTION DE LA SOLUTION

leurs modèles, le système devrait remédier à cette situation. Pour y parvenir, il faut fixer un seuil de temps égal au coût d'un tour C^t , qui représente le temps maximal nécessaire à un nœud pour entraîner et télécharger la mise à jour de son modèle sur la blockchain.

Après un nombre x de modèles validés, le premier agrégateur disponible les agrège dans un modèle intermédiaire. Le nombre de niveaux intermédiaires y est déterminé par la valeur de x qui est fixée en fonction du nombre et du type de noeuds dans le réseau et le nombre de modèles locaux attendu dans un tour FL. La figure 4.5 illustre un exemple d'agrégation avec $x = 3$.

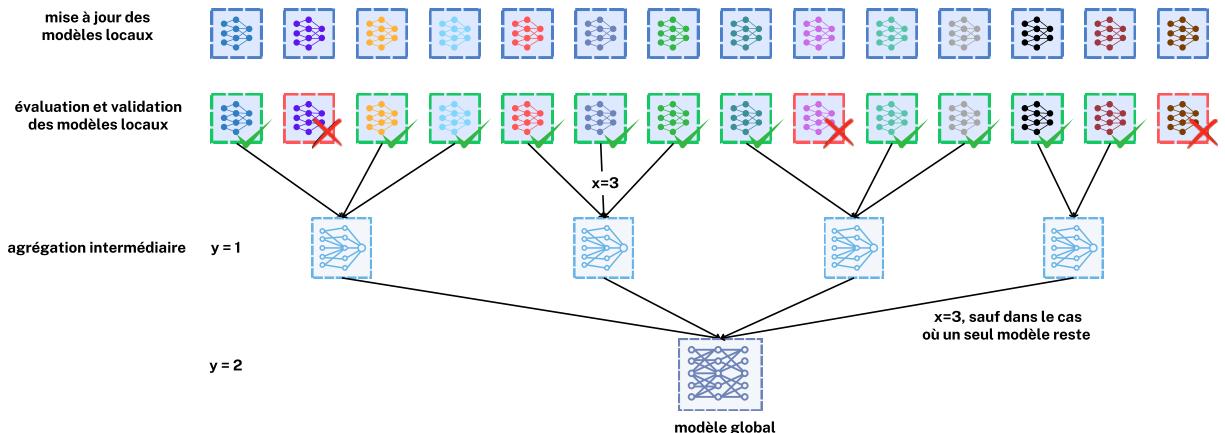


FIGURE 4.5 – Exemple d'agrégation.

Après le nombre nécessaire d'agrégations intermédiaires, l'agrégation globale finale, qui inclut tous les modèles locaux validés, sera effectuée et chargée sur la blockchain, marquant la fin de l'étape d'agrégation. Pour s'assurer que tous les modèles ont été agrégés, à chaque nouvelle agrégation intermédiaire qui est sauvegardée sur la blockchain, les modèles locaux ou bien intermédiaires inclus dans cette agrégation seront marqués comme agrégés. Lors de l'attribution des modèles à agréger, la priorité est toujours donnée aux modèles du niveau le plus bas, ainsi il est assuré que le niveau d'agrégation $y + 1$ ne commencera pas tant que tous les modèles du niveau $y - 1$ n'auront pas été agrégés ou bien en cours d'agrégation par d'autres au niveau y . Si le temps consommé par une évaluation ou une agrégation dépasse le seuil fixé, la tâche sera reprise par un autre agrégateur disponible.

La dernière étape d'un tour de FL consiste à calculer la contribution des nœuds, soit pour un apprentissage, soit pour les évaluations et les agrégations. Ensuite, ces valeurs de contributions déterminent le montant de la rémunération que les noeuds doivent recevoir et les scores d'honnêteté sont mis à jour. La valeur de rémunération du dispositif j dans un tour t est définie comme suit :

$$Rem_n^t = \frac{H_j^t}{\sum_{k \in K} H_k^t} \cdot \frac{\mathcal{B}}{T \cdot (K + J)} \quad (4.10)$$

Où K est le nombre de participants sélectionnés au tour t , J est le nombre d'agrégateurs sélectionnés au tour t et T le nombre de tours pour la tâche FL.

CHAPITRE 4. CONCEPTION DE LA SOLUTION

Une fois que le modèle global agrégé est chargé sur la blockchain, un contrat intelligent est déclenché pour effectuer cette tâche, en utilisant toutes les données nécessaires qui sont disponibles sur la blockchain. La figure 4.6 explique tout le processus d'agrégation à travers un diagramme de séquence.

4.5.2 Évaluation des modèles

Les modèles locaux

Afin de passer à l'étape d'agrégation un modèle doit passer par une évaluation afin de détecter s'il y eu une quelconque tentative de sabotage de la tâche FL. le nœud qui s'occupera de l'évaluation d'un modèle devra tester le modèle sur ses propres données et comparer entre la précision qu'il trouve et celle fournit avec le modèle par la part du nœud qui l'a entraîné. Soit $vote$ un vecteur à deux dimensions, la première composante représente le nombre de votes positifs et la deuxième composante le nombre de votes négatifs. L'algorithme 3 explique comment l'évaluation d'un modèle local est effectuée.

Algorithme 3 : Évaluation d'un modèle local

- 1 Étant donné un nœud agrégateur j , et un modèle local w_k avec une précision Acc_k .
 - 2 Test du modèle w_k sur les données locaux du noeud j et génération de Acc_j ,
 - 3 **if** $|Acc_k - Acc_j| \leq \phi$ **then**
 - 4 $vote_2 \leftarrow vote_2 + 1$
 - 5 **else**
 - 6 $vote_1 \leftarrow vote_1 + 1$
-

Les modèles intermédiaires

Les modèles intermédiaires sont le résultat du travail d'un nœud agrégateur sélectionné parmi les nœuds les plus honnêtes du réseau, ce qui permet d'évaluer leur travail d'une manière plus souple car l'évaluation des modèles intermédiaires n'est pas une exigence à la poursuite des étapes de l'agrégation. Ainsi les tâches d'agrégation serons plus prioritaires que celle de l'évaluation des modèles intermédiaires, donc les nœuds disponibles (après avoir affecter toutes les tâches d'agrégation) s'occuperont de ces évaluations qui consistent simplement à refaire l'agrégation et à vérifier que le modèle intermédiaire n'a pas été modifié, au cas où il l'aurait été, l'impact sur le score d'honnêteté du nœud qui a compromis le résultat sera conséquent.

4.6 Conclusion

Tout au long de ce chapitre, nous avons présenté notre solution. Nous avons pu voir comment notre approche résout les différents problèmes des systèmes d'apprentissage fédéré dans des environnements IoT. L'agrégation à plusieurs niveaux que nous proposons,

CHAPITRE 4. CONCEPTION DE LA SOLUTION

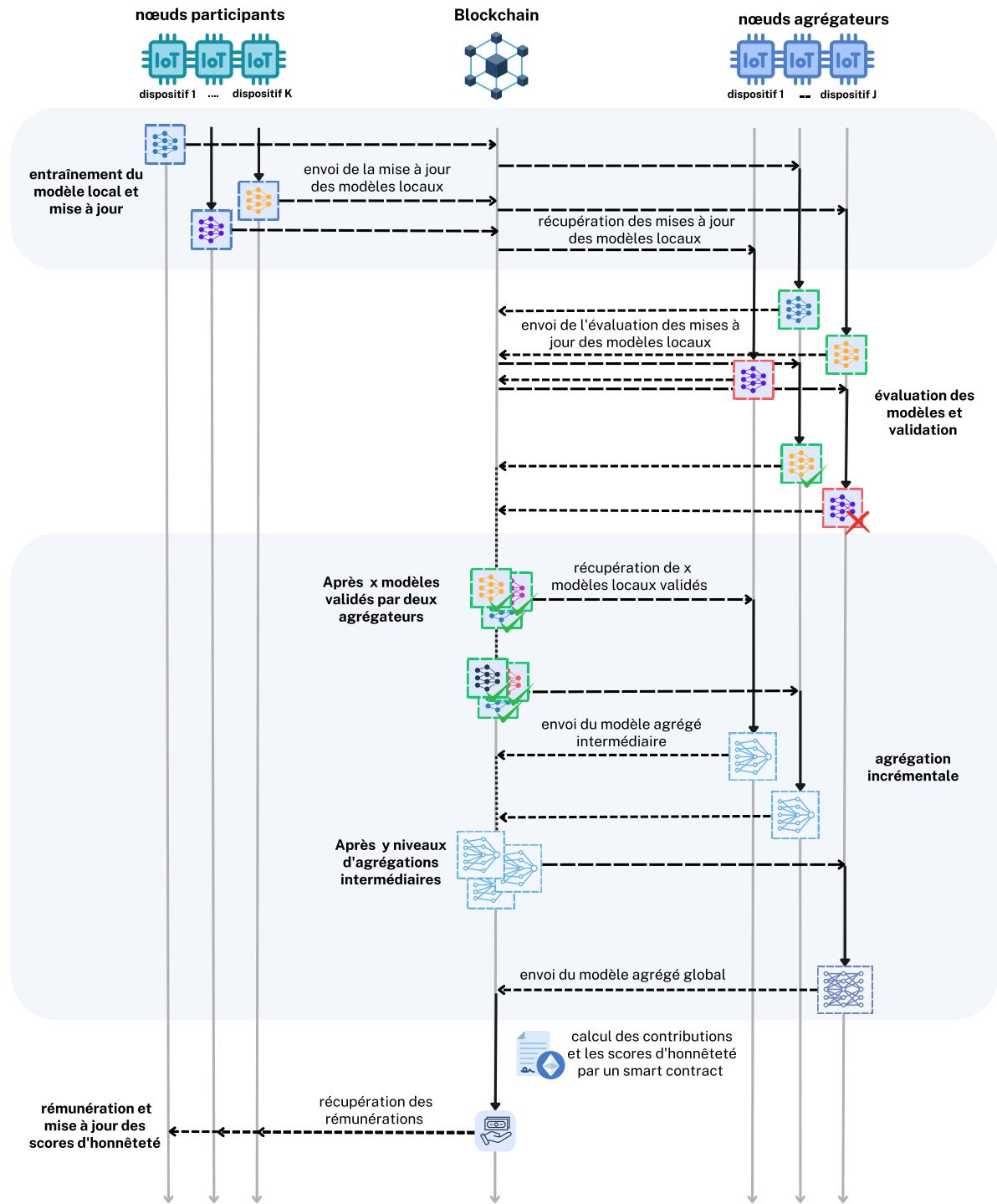


FIGURE 4.6 – Diagramme de séquence du processus d'évaluation et d'agrégation

CHAPITRE 4. CONCEPTION DE LA SOLUTION

résout le problème de point de défaillance unique ainsi que le besoin de synchronicité. En outre l'utilisation de la blockchain garantie la transparence et la fiabilité de notre système. Dans le chapitre qui va suivre nous allons implémenter notre solution et présenter en détails les résultats obtenus.

Chapitre 5

Réalisation et évaluation

5.1 Introduction

Le présent chapitre marque une étape cruciale dans notre travail de recherche, où nous nous plongeons dans la réalisation concrète de notre solution proposée. Après avoir défini les bases conceptuelles et théoriques dans les chapitres précédents, nous sommes désormais prêts à mettre en œuvre notre architecture distribuée dans un environnement simulé. Notre objectif est de démontrer l'efficacité et la robustesse de notre approche en confrontant notre système aux scénarios les plus divers et en évaluant sa performance.

Dans ce qui suit, nous allons d'abord présenter les outils, les librairies et les frameworks utilisés, ensuite, nous allons voir en détails l'environnement de simulation que nous avons mis en place pour évaluer notre système, ainsi que le setup expérimental. Nous décrirons les différents scénarios que nous avons choisis pour tester la solution proposée, et pour comparer entre les trois mécanismes de sélection mis en place.

5.2 Présentation des outils utilisés

5.2.1 Langages de programmation

Python

Python, créé en 1991 par Guido Van Rossum, est un langage de programmation de haut niveau qui allie approches orientées objet et fonctionnelles. Sa gestion automatique de la mémoire et son système robuste de gestion des exceptions assurent une expérience de développement stable. Sa syntaxe élégante favorise une programmation lisible. Son adoption massive en apprentissage automatique et science des données est due à ses bibliothèques riches facilitant des tâches complexes, de sites web dynamiques à la manipulation de données massives. Python se distingue par sa simplicité, puissance et adaptabilité, le rendant incontournable pour les développeurs mondiaux, particulièrement en apprentissage automatique et science des données.

C++

Le langage de programmation C++ est né dans les années 1980 grâce à Bjorne Stroustrup des laboratoires Bell. Évoluant à partir du langage C, il a introduit la programmation orientée objet, d'où son surnom de "C avec classes". En fusionnant ces concepts avec le langage C, C++ est devenu un outil polyvalent propice à une variété de développements logiciels, incluant navigateurs web avancés, simulateurs réseau et systèmes d'exploitation complets. Conservant des liens avec C, C++ combine haut niveau et accès bas niveau, lui conférant une position unique pour concilier abstraction flexible et manipulation précise des ressources matérielles.

5.2.2 Bibliothèques et framework

Numpy

NumPy est une bibliothèque essentielle pour Python, offrant des outils avancés de manipulation et d'analyse de données numériques. Elle simplifie les opérations mathématiques complexes, notamment les calculs matriciels et les statistiques avancées.

Pytorch

PyTorch, une bibliothèque majeure, est conçue pour le calcul tensoriel et le développement de réseaux neuronaux. Elle offre une plateforme flexible et performante pour la création de modèles d'apprentissage automatique. Grâce à sa structure dynamique de calcul, PyTorch simplifie la mise en place de prototypes et facilite le débogage.

GymAI

La bibliothèque GymAI est un atout essentiel pour le développement d'algorithmes d'apprentissage par renforcement. Elle offre une variété de simulations et de tâches qui simplifient la création, l'expérimentation et l'évaluation d'agents intelligents. Cette bibliothèque trouve une utilité cruciale dans la formation d'agents autonomes capables de prendre des décisions et de s'améliorer dans des environnements variés, allant des jeux aux systèmes complexes de contrôle.

Network Simulator NS3

Le simulateur NS-3 (Network Simulator 3) se présente comme un logiciel open-source de haute modularité, spécifiquement élaboré pour la simulation avancée des réseaux de communication. Il met à disposition une panoplie d'outils de modélisation de pointe, destinés à une évaluation minutieuse des performances, des protocoles et des comportements inhérents aux réseaux. En capitalisant sur une bibliothèque de modèles préétablis et une flexibilité notable, NS-3 permet aux chercheurs et développeurs d'élaborer des scénarios de simulation à forte fidélité, embrassant des topologies variées, allant des réseaux câblés aux environnements sans fil, et jusqu'aux réseaux véhiculaires. L'architecture modulaire qui le caractérise simplifie l'incorporation et la personnalisation de composants spécifiques

CHAPITRE 5. RÉALISATION ET ÉVALUATION

pour satisfaire des exigences particulières, tout en conformité avec les modèles de protocoles normalisés. En outre, pourvues de fonctionnalités de visualisation, doté d'un support script en langage Python, et ancré profondément dans le tissu de la recherche en réseaux, NS-3 demeure une ressource incontournable pour la validation conceptuelle, l'évaluation des performances, et la mise au point de protocoles novateurs. Cette influence est d'ailleurs attestée par un corpus conséquent de citations : NS-3 est cité dans plus de 25 500 travaux scientifiques, dont 2 640 se focalisent spécifiquement sur son application dans le contexte de la technologie Blockchain, et 520 se penchent sur son exploitation dans le domaine de l'apprentissage fédéré.

Bibliothèque ns3-ai

Le module "ns3-ai" est une extension pour le simulateur de réseaux ns-3, visant à faciliter l'interaction entre ns-3 et les frameworks d'intelligence artificielle basés sur Python. Il permet cette interaction en utilisant un pool de mémoire partagée pour le transfert de données entre les deux côtés, sans fournir d'algorithme d'IA ni dépendre de frameworks spécifiques. Contrairement à la bibliothèque déjà existante "ns3-gym", ce module offre une flexibilité accrue et une rapidité supérieure, élargissant l'échange de données au-delà de l'intégration de l'IA.

5.3 Détails de l'implémentation

5.3.1 Environnement de simulation

Avant d'entamer cette partie, nous tenons à préciser les caractéristiques de la machine utilisée pour tester l'approche proposée :

— Caractéristiques du CPU

```
(base) hiba@DESKTOP-DHCSIEB:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
Address sizes:         39 bits physical, 48 bits virtual
CPU(s):                4
On-line CPU(s) list:  0-3
Thread(s) per core:   2
Core(s) per socket:   2
Socket(s):             1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 78
Model name:            Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz
Stepping:               3
CPU MHz:                2495.998
BogoMIPS:               4991.99
Hypervisor vendor:     Microsoft
Virtualization type:   full
L1d cache:              64 KiB
L1i cache:              64 KiB
L2 cache:                512 KiB
L3 cache:                3 MiB
```

FIGURE 5.1 – Performance de la machine utilisée en terme de CPU

— Caractéristiques de la RAM

	total	used	free	shared	buff/cache	available
Mem:	12	11	0	0	0	0
Swap:	4	0	3			

FIGURE 5.2 – Performance de la machine utilisée en terme de RAM

Afin de démontrer l’efficacité inhérente à notre approche, nous avons adopté plusieurs scénarios de test au sein d’un même environnement simulé au moyen du simulateur NS3¹. Cet environnement est composé d’un ensemble de noeuds IoT, définis par les caractéristiques suivantes : leur disponibilité, la taille des données associées à la tâche d’apprentissage fédéré, la fréquence CPU, le débit de transmission et un indice d’honnêteté. Leurs valeurs ont été attribuées de manière aléatoire, ce qui nous permet de simuler l’hétérogénéité entre ces noeuds. Par exemple, la taille des données varie entre 50 et 1000 instances, la fréquence varie entre 50 et 300 MHz, et le délai de transmission fluctue entre 150 et 1000 Mbps.

En ce qui concerne la disponibilité des noeuds, tous les scénarios sont simulés avec une disponibilité de 80 % tout au long du processus d’apprentissage fédéré. Les indices d’honnêteté sont initialement définis à zéro, puis évoluent en fonction du comportement et de l’intégrité démontrés par chaque noeud. En plus des noeuds IoT, chargés de la réalisation des tâches d’apprentissage fédéré, le réseau comprend également des noeuds blockchain. Ces derniers prennent en charge les aspects liés à la blockchain, tels que l’enregistrement des transactions, la sélection des participants au processus d’apprentissage fédéré, ainsi que la planification des évaluations et de l’agrégation des modèles.

Un noeud supplémentaire représente l’entité responsable du déclenchement de la tâche d’apprentissage fédéré, également connu sous le nom d’initiateur. Tous ces noeuds sont interconnectés et ont la capacité de communiquer au sein d’un réseau Wi-Fi de type ad hoc. Ce dernier est utilisé dans les réseaux IoT en raison de leur nature décentralisée, de leurs faibles exigences en matière d’infrastructure, de leur flexibilité et de leur pertinence pour les scénarios où une communication directe entre appareils est avantageuse. Ils permettent aux dispositifs IoT d’établir des connexions locales sans dépendre d’une infrastructure réseau étendue, ce qui les rend utiles dans les zones éloignées ou à infrastructure limitée.

5.3.2 Architecture des réseaux de neurones et dataset utilisés

Entraînement local

Le modèle de réseau de neurones que nous employons est simple mais efficace. Il comprend deux couches linéaires (entièrement connectées). La première couche prend en entrée les images, préalablement aplatis en un vecteur de 784 dimensions (28x28), et la

1. <https://github.com/PFEWorkspace/code.git>

CHAPITRE 5. RÉALISATION ET ÉVALUATION

seconde génère des scores pour la classification dans les 10 classes de chiffres. La fonction d'activation ReLU est intercalée entre ces couches pour introduire de la non-linéarité.

Nous utilisons la méthode de minimisation de la perte "CrossEntropyLoss" associée à l'optimiseur "Adam" pour entraîner notre modèle. Les paramètres du modèle, tels que la taille du lot (batch size), le taux d'apprentissage (learning rate) et le nombre d'époques (epochs), ont été définis respectivement à 64, 0.001 et 5 pour tout les nœuds et les différents scénarios.

Apprentissage par renforcement profond

Concernant le DRL nous avons choisi d'utiliser un Soft Actor Critic, l'agent est donc composé de trois types de réseaux de neurones tous avec les mêmes couches cachées (2 couches de dimension 256) ainsi que d'un ReplayBuffer :

- **Actor Network** : il est représenté par un réseau de neurones qui prend en entrée l'observation de l'environnement soit un objet de dimension (nombre de nœuds, nombre d'attributs), dans notre cas le nombre de nœuds diffère selon les scénarios et le nombre d'attributs est fixe (8), en sortie nous avons l'action soit la liste des indices des nœuds sélectionnées, encore une fois les dimensions dépendent du scénario d'exécution.
- **Critic Network** : il est représenté par un réseau de neurones qui prend en entrée une observation ainsi qu'une action, en sortie nous avons la Q-value soit un objet de dimension 1.
- **Value Network** : il est représenté par un réseau de neurones qui prend en entrée une observation, en sortie nous avons la V-value soit un objet de dimension 1.
- **ReplayBuffer** : il représente une structure qui permet de sauvegarder les transition c.a.d l'état de l'environnement avant et après l'exécution de l'action ainsi que la récompense et l'état de la tache FL (état terminal ou non) liés à cette action

Les différents réseaux utilisent le même optimiseur "Adam" ainsi que la même fonction d'activation ReLu.

5.3.3 Paramétrage

Dans cette section, nous examinons les différents paramètres employés dans les scénarios de simulation. Ces paramètres jouent un rôle essentiel dans l'évaluation de l'efficacité de notre solution dans diverses situations. Le tableau 5.1 présente les paramètres utilisés, ainsi que les valeurs qui leur sont associées.

CHAPITRE 5. RÉALISATION ET ÉVALUATION

TABLE 5.1 – Paramètres de simulation

Paramètres	Description
nodes	Le nombre de nœuds FL dans le réseau, ces valeurs de simulation = {50, 150, 300}
trainers	Le nombre de noeuds participants (pour l'entraînement local) à sélectionner {20, 30, 80}
aggregators	Le nombre de nœuds évaluateurs/aggrégateurs à sélectionner {10, 20, 50}
BC_nodes	Le nombre de nœuds blockchain dans le réseau {30, 50, 100}
x	Le nombre de modèles locaux/intermédiaires à agréger à la fois {5, 6, 12}
selection	L'approche de sélection à utiliser {score, DRL, hybrid}
α	Le facteur α de la formule du score, afin de compromettre entre l'honnêteté et le coût {0.2, 0.5, 0.8}
dropout	Le pourcentage des nœuds qui abandonnent leurs tâches {10%, 30%, 60%}
malicious	Le pourcentage des nœuds malhonnêtes/compromis {10%, 30%, 60%}

La variation du nombre de nœuds permet de mener une comparaison des performances des trois approches sur différentes échelles, offrant ainsi un aperçu de la scalabilité relative de ces solutions. Ensuite, en faisant varier la valeur du facteur α , il devient possible d'analyser l'influence du compromis entre l'honnêteté et le coûts sur le processus de sélection par score et, en fin de compte, sur la convergence de l'apprentissage fédéré. D'autre part, le pourcentage d'abondant (dropout) permet d'évaluer la disponibilité du système et la tolérance vis-à-vis des abondants, témoignant ainsi de l'efficacité de l'agrégation à plusieurs niveaux proposée, et aussi l'algorithme de sélection qui va permettre, ou non, de détecter ces cas. De même, en ajustant le pourcentage de nœuds malveillants, nous mettons en évidence la capacité du système à résister aux attaques d'empoisonnement et aux activités compromises, ce qui confirme son niveau de sécurité et de confiance.

TABLE 5.2 – Autres paramètres de Simulation

Paramètre	Valeur
Pourcentage de la partition de test	20%
Pourcentage de disponibilité des nœuds	80%
Nombre de tours FL	30
Objectif en termes de précision du modèle global	99%
Nombre d'epochs pour l'entraînement local	5
Taille du lot pour l'entraînement local	64
Paramètres de la formule d'honnêteté	$\alpha = 1, \beta = 3, \gamma = 10, \phi = 2$
Pénalité d'abandon	-10
Seuil de validation d'un modèle local	7

CHAPITRE 5. RÉALISATION ET ÉVALUATION

Parmi les autres paramètres employés, leurs valeurs restent constantes pour l'ensemble des scénarios. Cette démarche vise à garantir que les trois approches proposées sont évaluées dans des conditions identiques, permettant ainsi des comparaisons équitables. De plus, cela inclut des éléments qui se rapprochent du monde réel, tels que le pourcentage de nœuds disponibles, fixé à 80 %, et les valeurs des différents facteurs utilisés dans les formules et algorithmes proposées dans la contribution, ces derniers sont précisés dans le tableau 5.2

5.4 Évaluation

Cette section englobe une évaluation approfondie de notre approche, prenant en compte les trois mécanismes de sélection. L'objectif ultime est de les comparer exhaustivement. Notre évaluation se concentre principalement sur deux indicateurs clés : la convergence du modèle global dans le contexte de l'apprentissage fédéré, quantifiée par la précision du modèle en fonction du nombre de rounds FL, et le temps d'exécution (le temps de simulation, qui est indépendant de la machine). Ces indicateurs permettent d'apprécier si les mécanismes de sélection réussissent à identifier les nœuds offrant les meilleures performances (convergence optimale) et à moindre coût. En plus de cela, nous analysons également l'évolution du score d'honnêteté au fil du temps pour les différentes approches proposées.

La suite de cette section présente une série de scénarios de simulation ainsi que les résultats obtenus. Ces scénarios sont organisés en fonction des mécanismes de sélection. Pour chaque approche de sélection, trois scénarios distincts sont envisagés. Le premier scénario, plus élémentaire, vise à évaluer la mise en échelle : les performances des approches proposées en considérant diverses tailles de réseau. L'objectif est de déterminer la capacité des approches à être scalables. Le deuxième scénario implique une variation du pourcentage de nœuds malveillants, visant à vérifier si le système est résistant face à leurs attaques et si les mécanismes de sélection sont capables de les éliminer. Enfin, le dernier scénario examine la variation du pourcentage d'abandon (dropout), afin de déterminer si le système parvient à maintenir son niveau de disponibilité et si les mécanismes de sélection sont aptes à anticiper ce type de comportement des nœuds.

5.4.1 Sélection basée sur le score

— Scénario 1

TABLE 5.3 – Paramètres du scénario 1 pour la sélection basée sur le score

nodes	trainers	aggregators	BC_nodes	x	α	dropout	malicious
50	20	10	30	5	$\alpha = 0.2, 0.5, 0.8$	10%	10%
150	30	20	50	6	$\alpha = 0.2, 0.5, 0.8$	10%	10%
300	80	50	100	12	$\alpha = 0.2, 0.5, 0.8$	10%	10%

CHAPITRE 5. RÉALISATION ET ÉVALUATION

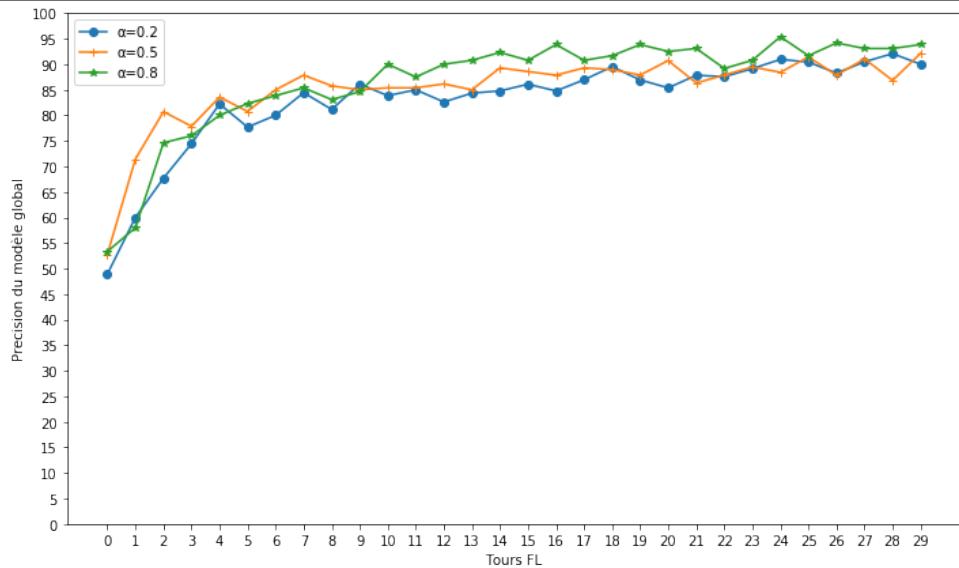


FIGURE 5.3 – Convergence du modèle global avec sélection basée sur le score avec 50 noeuds et variation du paramètre α

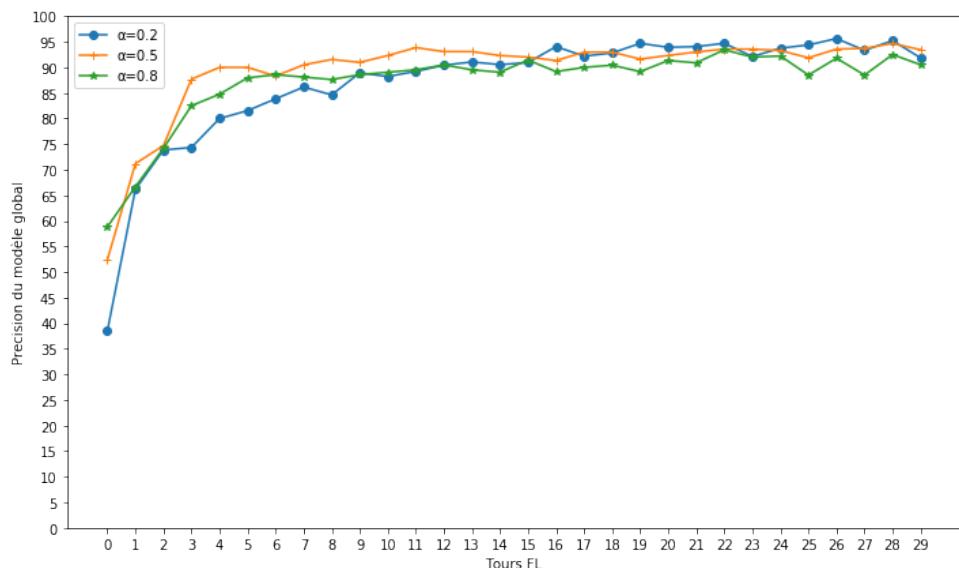


FIGURE 5.4 – Convergence du modèle global avec sélection basée sur le score avec 150 noeuds et variation du paramètre α

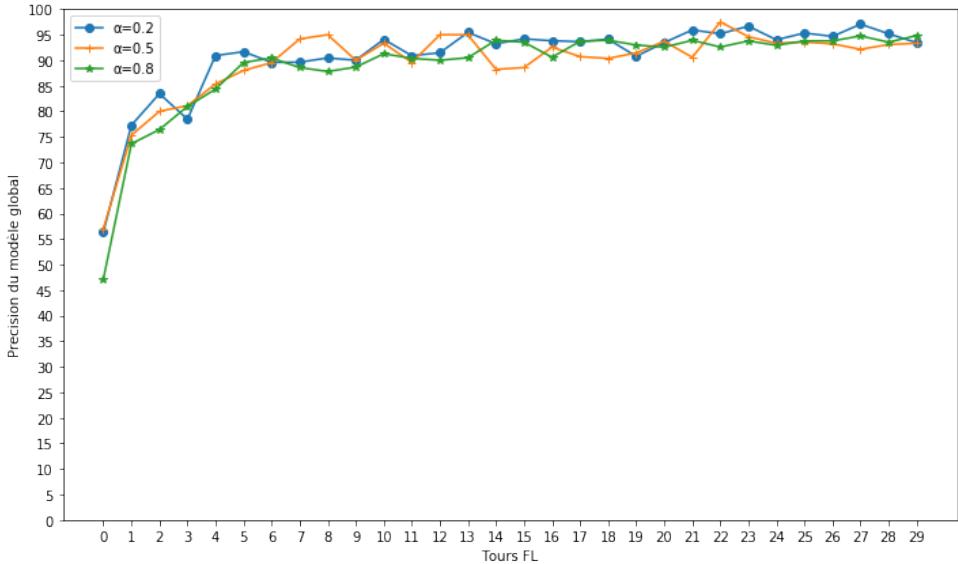


FIGURE 5.5 – Convergence du modèle global avec sélection basée sur le score avec 300 noeuds et variation du paramètre α

Nombre de noeuds	Valeurs de α		
	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$
nodes=50	90.0%	92.14%	93.85%
nodes=150	91.84%	93.43%	90.5%
nodes=300	93.44%	93.33%	94.83%

TABLE 5.4 – La précision finale du modèle global du scénario 1 avec sélection basée sur le score

Avec les paramètres précisés dans le tableau 5.3, les résultats obtenus sont exposés dans le tableau 5.4 et les figures 5.3, 5.4 et 5.5. Pour le scénario comportant 50 noeuds, et ce, pour les trois valeurs de α , à savoir 0.2, 0.5 et 0.8, les taux de précision finaux du modèle global s'établissent respectivement à 90.0%, 92.14% et 93.85%. En ce qui concerne le scénario à 150 noeuds, les taux de précision globaux sont de 91.84%, 93.43% et 90.5% pour les mêmes valeurs de α . Enfin, dans le contexte de 300 noeuds, les résultats sont les suivants : 93.44%, 93.33% et 94.83%.

Les figures 5.3, 5.4 et 5.5 mettent en évidence une observation intéressante : la valeur du paramètre α dépend de la communauté de noeuds et sa taille, de leurs caractéristiques ainsi que le pourcentage de sélection. Par exemple, pour le cas de 50 noeuds où nous avons 60% de noeuds sélectionnés par rapport au réseau, illustré par la figure 5.3, la sélection basée sur le score avec α à 0.8 a été la plus performante avec une convergence du modèle global plus élevée et une meilleure stabilité. Cette dernière découle du fait que l'accent est davantage mis sur l'honnêteté. Par ailleurs, on peut voir sur la figure 5.4, où nous avons 150 noeuds avec un pourcentage de sélection à 33%, que la sélection avec $\alpha = 0.5$ est plus adéquate à ce scénario. En

CHAPITRE 5. RÉALISATION ET ÉVALUATION

général, quelle que soit la valeur des paramètres impliqués, la sélection basée sur le score s'est avérée assez performante pour produire des résultats satisfaisants.

La figure 5.6 présente l'évolution des valeurs du score d'honnêteté des noeuds au fil des tours de l'apprentissage fédéré, dans un scénario comprenant 50 noeuds avec $\alpha = 0.5$. Cette visualisation met en lumière la trajectoire temporelle des score d'honnêteté pour chaque noeud, révélant ainsi leur évolution. La formule du score d'honnêteté repose sur la contribution de chaque noeud, mais tient également compte du comportement d'un noeud, entraînant ainsi une diminution de son score d'honnêteté lors d'un acte malveillant. Cette réduction a pour effet de rendre le noeud moins susceptible d'être sélectionné par la suite, sauf en cas de rareté de noeuds honnêtes. L'impact de la malveillance sur le score d'honnêteté est clairement visible, contribuant à la prise de décision quant à l'inclusion ou à l'exclusion d'un noeud dans le processus de sélection.

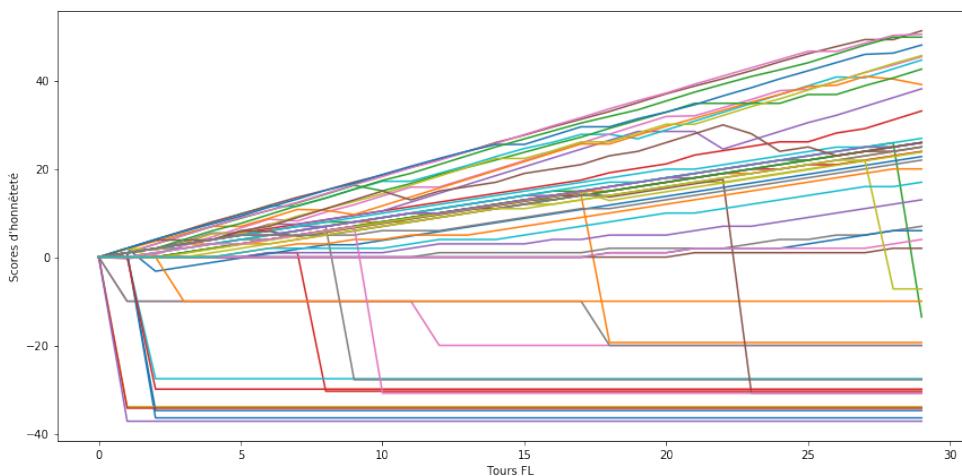


FIGURE 5.6 – Les valeurs du score d'honnêteté des noeuds avec sélection basée sur le score.

— Scénario 2

TABLE 5.5 – Paramètres du scénario 2 pour la sélection basée sur le score

nodes	trainers	aggregators	BC_nodes	x	α	dropout	malicious
50	20	10	30	5	$\alpha = 0.5$	10%	10% 30% 60%

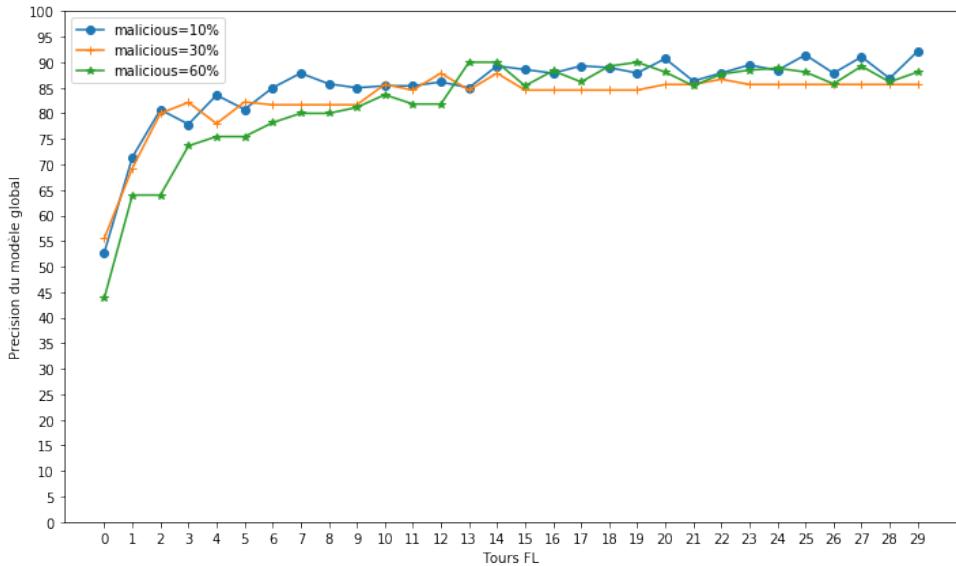


FIGURE 5.7 – Convergence du modèle global avec sélection basée sur le score avec différents pourcentages de nœuds malveillants

malicious	précision
10%	92.14%
30%	85.65%
60%	88.08%

TABLE 5.6 – La précision finale du modèle global du scénario 2 avec sélection basée sur le score

Le scénario adoptant les paramètres répertoriés dans le tableau 5.5, et variant le pourcentage de nœuds malveillants au sein du réseau, a engendré les résultats exposés dans le tableau 5.6 et la figure 5.7. Comme l'illustrent les données, le système démontre une résilience adéquate face aux attaques d'empoisonnement, préservant ainsi les performances du modèle à des niveaux élevés de qualité, lesquels se traduisent par des taux de précision établis à 92.14%, 85.65% et 88.08%. Ces pourcentages correspondent respectivement aux proportions de nœuds malveillants de 10%, 30% et 60%.

Cet aspect positif peut être attribué au mécanisme d'évaluation et de détection de modèles empoisonnés qui a été mis en place, mais aussi au mécanisme de sélection qui élimine les nœuds malveillants, permettant au système de maintenir ses performances malgré la présence d'acteurs malveillants dans le réseau.

— Scénario 3

TABLE 5.7 – Paramètres du scénario 3 pour la sélection basée sur le score

nodes	trainers	aggregators	BC_nodes	x	α	dropout	malicious
50	20	10	30	5	$\alpha = 0.5$	10% 30% 60%	10%

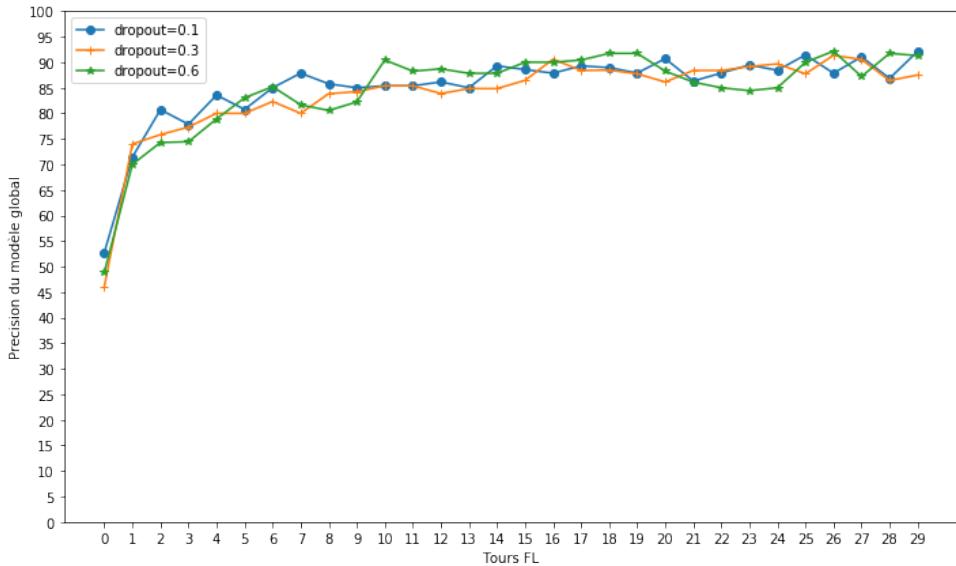


FIGURE 5.8 – Convergence du modèle global avec sélection basée sur le score avec différents pourcentages d'abandon

dropout	précision
10%	92.14%
30%	87.5%
60%	91.3%

TABLE 5.8 – La précision finale du modèle global du scénario 3 avec sélection basée sur le score

Dans le contexte de ce scénario, qui est défini par le tableau 5.7, les résultats obtenus sont exposés dans le tableau 5.8 et la figure 5.8. Ces résultats sont quantifiés à des taux de précision de 92.14%, 87.5% et 91.3%, correspondant à des proportions d'abandon de tâches parmi les nœuds de 10%, 30% et 60%. Une observation importante réside dans le maintien des performances de l'apprentissage fédéré, qui ne connaît pas une chute significative, préservant ainsi un niveau de convergence remarquablement élevé.

Cette résilience est attribuée au mécanisme de sélection par score d'honnêteté ainsi qu'à l'agrégation multi-niveaux mis en œuvre. Ce mécanisme permet de détecter les nœuds qui tendent à abandonner fréquemment leurs tâches, et par conséquent, de les pénaliser au fil du temps, aboutissant à leur non-sélection progressive. Cependant, il est pertinent de noter qu'avec un taux d'abandon de 60%, le système est contraint de choisir parmi ces nœuds. Cette situation se traduit par une diminution substantielle du nombre de modèles locaux générés par les nœuds, du fait de leurs abandons répétés de tâches et de leur omission de soumettre des modèles. Par conséquent, les performances du modèle global sont compromises, à cause du manque de modèles locaux disponibles.

5.4.2 Sélection basée sur le DRL

Dans cette partie, nous allons effectuer nos tests avec un agent DRL sans pré-entraînement.

— Scénario 1

TABLE 5.9 – Paramètres du scénario 1 pour la sélection DRL

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	20	10	30	5	10%	10%
150	30	20	50	6	10%	10%
300	80	50	100	12	10%	10%

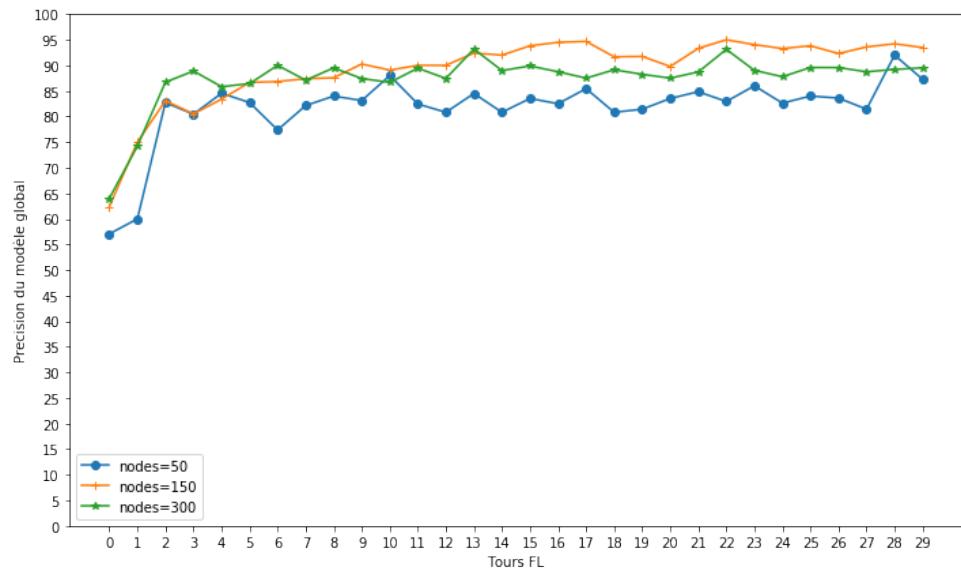


FIGURE 5.9 – Convergence du modèle global avec sélection DRL avec différents nombres de noeuds.

nodes	précision
50	85.71%
150	93.47%
300	89.62%

TABLE 5.10 – La précision finale du modèle global du scénario 1 avec sélection par DRL

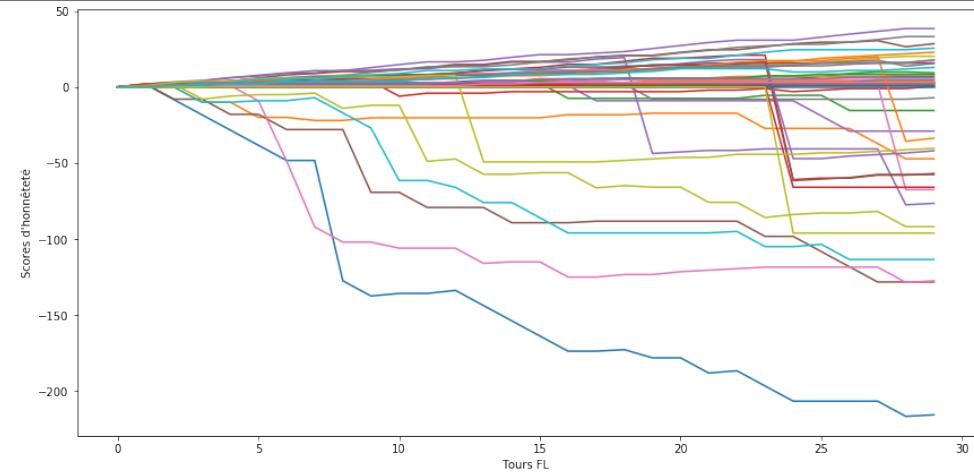


FIGURE 5.10 – Les valeurs du score d’honnêteté des noeuds avec sélection DRL.

Les résultats obtenus de ce scénario, exposés en détails dans le tableau 5.10 et illustrés par la figure 5.9, se traduisent par des précisions finales du modèle global de 85.71%, 93.47% et 89.62% respectivement pour 50, 150 et 300 noeuds. Ces valeurs s’avèrent acceptables, étant donné qu’un nombre adéquat de tours de l’apprentissage fédéré a été effectué pour atteindre ces résultats. Toutefois, même en prenant en compte des pourcentages d’abandons et de noeuds malhonnêtes de 10% chacun, la figure 5.9 met en évidence une instabilité dans la convergence du modèle. Cette instabilité peut être attribuée à un mauvais choix des noeuds et à une négligence de leurs degrés d’honnêteté, ainsi qu’à la qualité de leurs caractéristiques (telles que la taille du jeu de données, par exemple).

Cette situation est clairement démontrée par la figure 5.10, qui expose les variations des valeurs d’honnêteté dans le scénario avec 50 noeuds. Il est à noter que même lorsque le niveau d’honnêteté d’un noeud se trouve déjà en territoire négatif, l’agent DRL continue de le sélectionner. Cette tendance souligne la nécessité d’améliorer la capacité de l’agent DRL à évaluer les noeuds en fonction de leur honnêteté et de leurs caractéristiques pour assurer une convergence plus stable du modèle global.

— Scénario 2

TABLE 5.11 – Paramètres du scénario 2 pour la sélection DRL

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	20	10	30	5	10%	10% 30% 60%

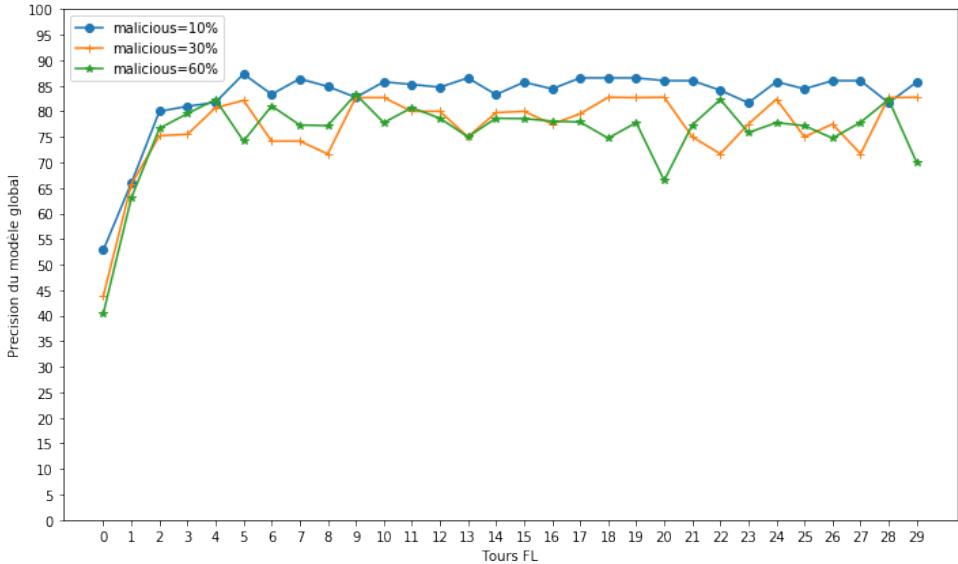


FIGURE 5.11 – Convergence du modèle global avec sélection DRL avec différents pourcentages de noeuds malveillants.

malicious	précision
10%	85.71%
30%	82.75%
60%	70.0%

TABLE 5.12 – La précision finale du modèle global du scénario 2 avec sélection par DRL

Les résultats issus de ce scénario, caractérisé par différents taux de noeuds malhonnêtes au sein du réseau conformément au tableau 5.11, sont représentés de manière graphique dans le tableau 5.12 et la figure 5.11. En ce qui concerne la précision finale du modèle global, celle-ci s'établit à 85.71%, 82.75% et 70.0% respectivement pour des taux de noeuds malveillants de 10%, 30% et 60%. Ces résultats montrent une dégradation des performances de l'apprentissage fédéré, à travers une fluctuation significative dans la convergence du modèle global, cette dernière augmente en fonction du taux de noeuds malhonnêtes. Cette variabilité découle de l'incapacité de l'agent DRL à anticiper le comportement des noeuds malveillants et ainsi protéger le système contre leurs attaques d'empoisonnement.

— Scénario 3

TABLE 5.13 – Paramètres du scénario 3 pour la sélection DRL

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	20	10	30	5	10% 30% 60% s	10%

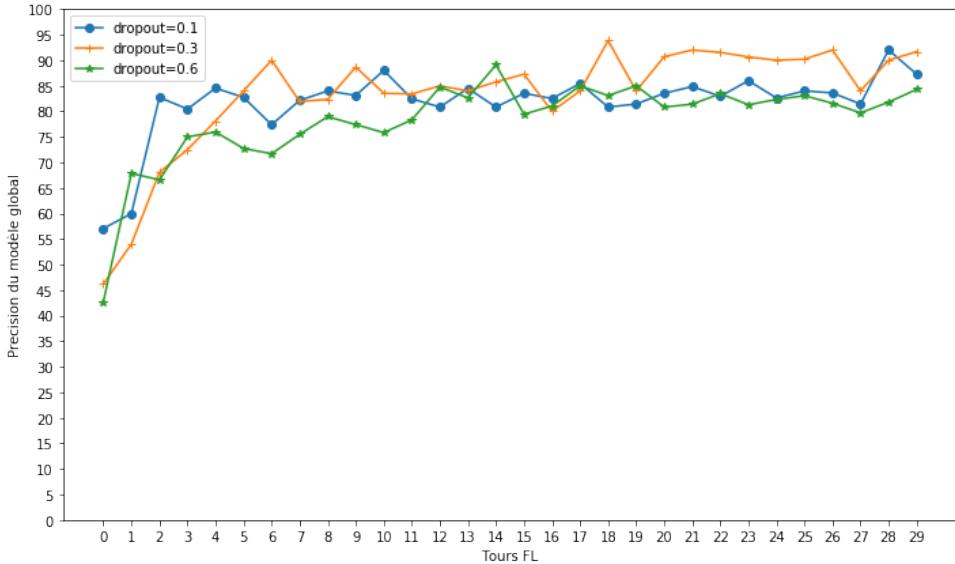


FIGURE 5.12 – Convergence du modèle global avec sélection DRL avec différents pourcentages d’abandon.

dropout	précision
10%	87.27%
30%	91.67%
60%	84.29%

TABLE 5.14 – La précision finale du modèle global du scénario 3 avec sélection par DRL

Ce scénario est décrit dans le tableau 5.13 et ses résultats sont exposés dans la figure 5.12. La précision finale du modèle global se chiffre à 87.27%, 91.67% et 84.29% respectivement, pour des taux d’abandon parmi les noeuds de 10%, 30% et 60%. Selon la figure 5.12, il est observable que plus le pourcentage d’abandons augmente, plus la précision du modèle diminue en raison de la réduction du nombre de modèles locaux à agréger.

5.4.3 Sélection hybride

La sélection hybride est précédée de deux étapes, la première étant les exécutions des différents scénarios avec la méthode exacte, depuis ces exécutions nous sauvegardons dans le ReplayBuffer de notre agent chaque transition effectuée soit un observation, l'action choisie, l'observation qui en découle, la récompense ainsi que le flag done (indiquant la fin de la tache FL). Dans la deuxième étape nous allons entraîner notre agent sur ces données jusqu'à stabilisation avant de le déployer pour utilisation.

Les résultats présentés dans cette section sont issus des exécutions avec sélection basée sur l’agent DRL entraîné.

— Scénario 1

CHAPITRE 5. RÉALISATION ET ÉVALUATION

TABLE 5.15 – Paramètres du scénario 1 pour la sélection hybride

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	20	10	30	5	10%	10%
150	30	20	50	6	10%	10%
300	80	50	100	12	10%	10%

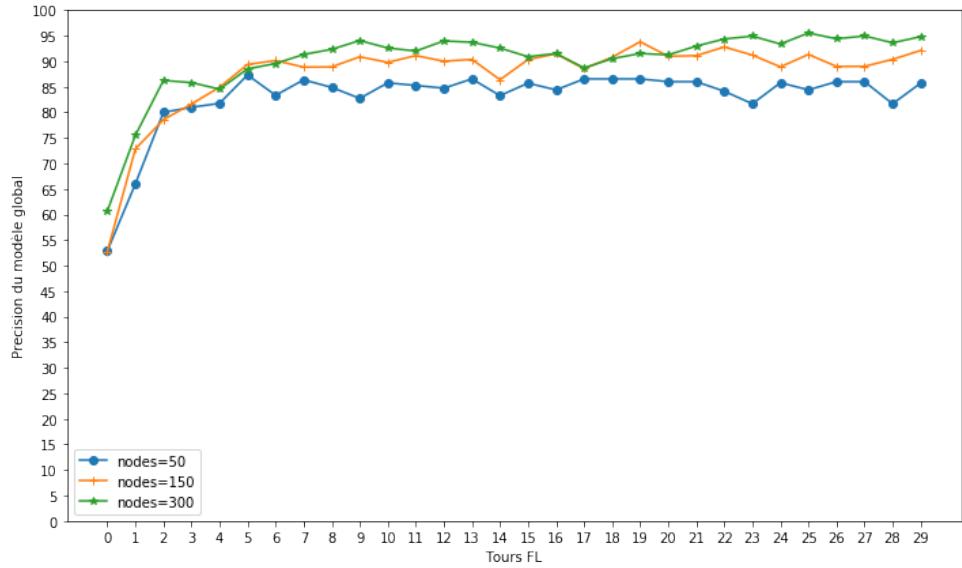


FIGURE 5.13 – Convergence du modèle global avec sélection hybride avec différents nombres de noeuds.

nodes	précision
50	85.71%
150	92.11%
300	94.72%

TABLE 5.16 – La précision finale du modèle global du scénario 1 avec sélection hybride

La figure 5.13 ainsi que le tableau 5.16 montrent les résultats du scénario d'exécution décrit par le tableau 5.15 pour la méthode de sélection hybride. Nous nous retrouvons ainsi avec des précisions finales du modèle global de 85.71%, 92.11% et 94.72% respectivement pour 50, 150 et 300 noeuds. Au sein de cette figure, nous remarquons que la méthode hybride atteint rapidement un bon niveau de convergence et maintient une certaine stabilité, cela est dû au fait que l'agent utilisé a été pré-entraîné ce qui lui a permis de prédire quels sont les noeuds susceptibles d'améliorer la convergence du modèle et d'éviter les noeuds malveillants.

Nous pouvons voir sur la figure 5.14, qui représente les scores d'honnêteté des noeuds, que parfois l'agent sélectionne des noeuds ayant déjà des scores d'honnêteté négatifs

CHAPITRE 5. RÉALISATION ET ÉVALUATION

cela est dû au fait qu'il soit encouragé à explorer le spectre d'actions qui s'offre à lui, mais il réussit quand même à détecter la plupart des nœuds malhonnêtes et ne les choisit plus. Ils sont représentés dans la figure par les nœuds dont l'honnêteté diminue une fois et reste stable pour le reste de la tâche ce qui prouve qu'ils ne sont plus sélectionnés.

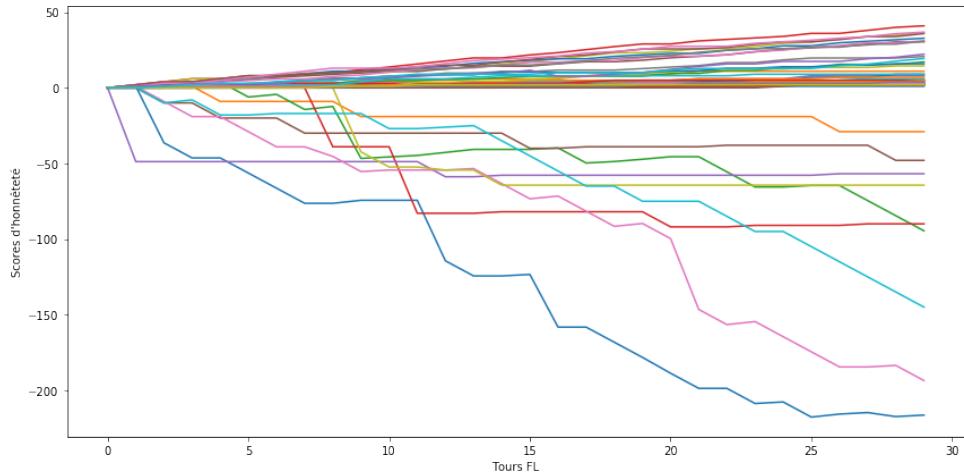


FIGURE 5.14 – Les valeurs du score d'honnêteté des nœuds avec sélection hybride.

— Scénario 2

TABLE 5.17 – Paramètres du scénario 2 pour la sélection hybride

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	20	10	30	5	10%	10% 30% 60%

malicious	précision
10%	85.71%
30%	88.46%
60%	87.31%

TABLE 5.18 – La précision finale du modèle global du scénario 2 avec sélection hybride

La figure 5.15 illustre la progression de la convergence du modèle global selon le scénario d'exécution détaillé dans le tableau 5.17. Nous observons une précision accrue du modèle, atteignant 85.71%, 88.46% et 87.31% pour les réseaux comprenant 10%, 30% et 60% de nœuds malveillants respectivement. L'analyse de la Figure 5.15 révèle que l'approche hybride détecte efficacement les nœuds malveillants et les exclut de ses choix. Cette observation est étayée par la stabilité de la convergence du modèle, qui ne présente pas de fluctuations marquées. De plus, la performance de cette approche ne se dégrade pas beaucoup lorsque le taux de malhonnêteté augmente.

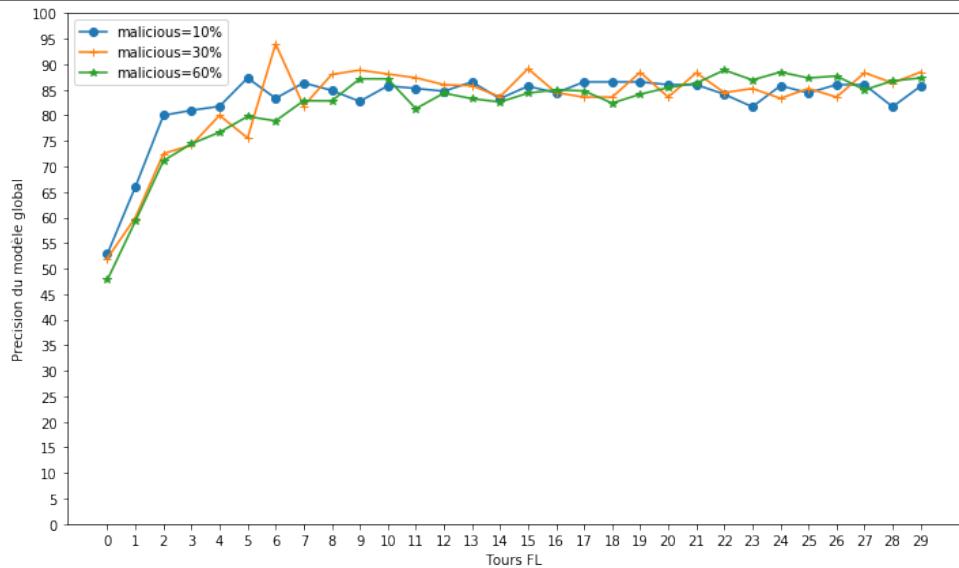


FIGURE 5.15 – Convergence du modèle global avec sélection hybride avec différents pourcentages de nœuds malveillants.

— Scénario 3

TABLE 5.19 – Paramètres du scénario 3 pour la sélection hybride.

nodes	trainers	aggregators	BC_nodes	x	dropout	malicious
50	30	20	30	5	10% 30% 60%	10%

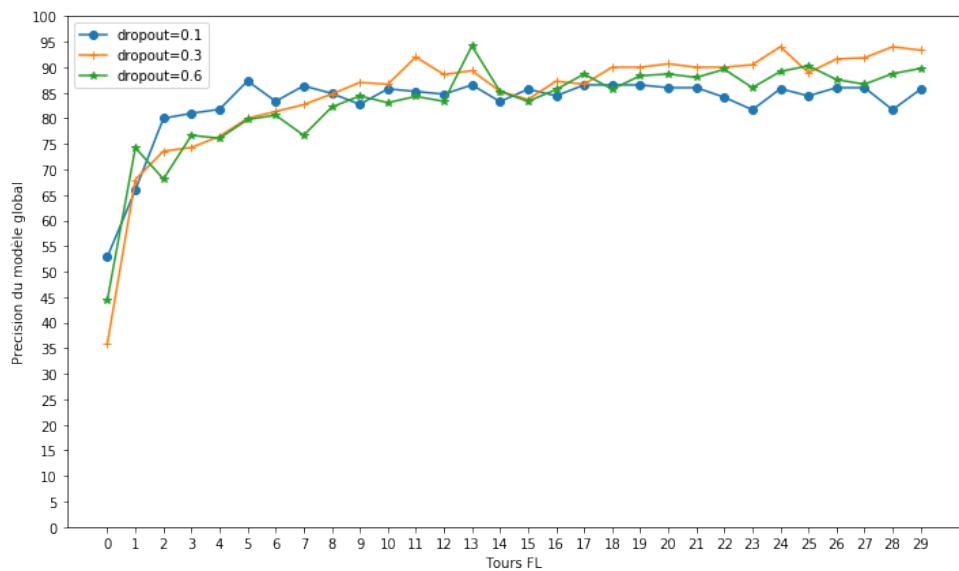


FIGURE 5.16 – Convergence du modèle global avec sélection hybride avec différents pourcentages d'abandon.

dropout	précision
10%	85.7%
30%	93.3%
60%	89.75%

TABLE 5.20 – La précision finale du modèle global du scénario 3 avec sélection hybride

Ce scénario est détaillé dans le tableau 5.19, et les résultats qui en découlent sont présentés dans la figure 5.16 et le tableau 5.20. Les performances finales du modèle global se traduisent par des taux de précision respectifs de 85.7%, 93.3% et 89.75% pour des niveaux d'abandon de noeuds atteignant 10%, 30% et 60%. En analysant la figure 5.16, il est perceptible que même lorsque le pourcentage d'abandons augmente la précision du modèle global reste acceptable.

5.4.4 Comparaison et Synthèse

Après avoir examiné les résultats de chaque approche de manière indépendante, pour chaque scénario proposé, nous entreprenons désormais une comparaison afin de déterminer si cette hybridation engendre de meilleurs résultats.

À cette fin, nous proposons une évaluation comparative des résultats obtenus pour chaque approche de sélection, en nous concentrant sur un cas singulier pour chaque scénario (le cas le plus extrême). Cette analyse comparative comporte le cas avec un réseau de 300 noeuds, correspondant au premier scénario, comme illustré dans la figure 5.17. Pour le deuxième scénario, nous considérons spécifiquement le cas où le pourcentage de noeuds malhonnêtes est de 60%, comme présenté dans la figure 5.18. Enfin, pour le dernier scénario, nous nous penchons sur le cas où le taux d'abandons atteint 60%, tel qu'exemplifié dans la figure 5.19.

CHAPITRE 5. RÉALISATION ET ÉVALUATION

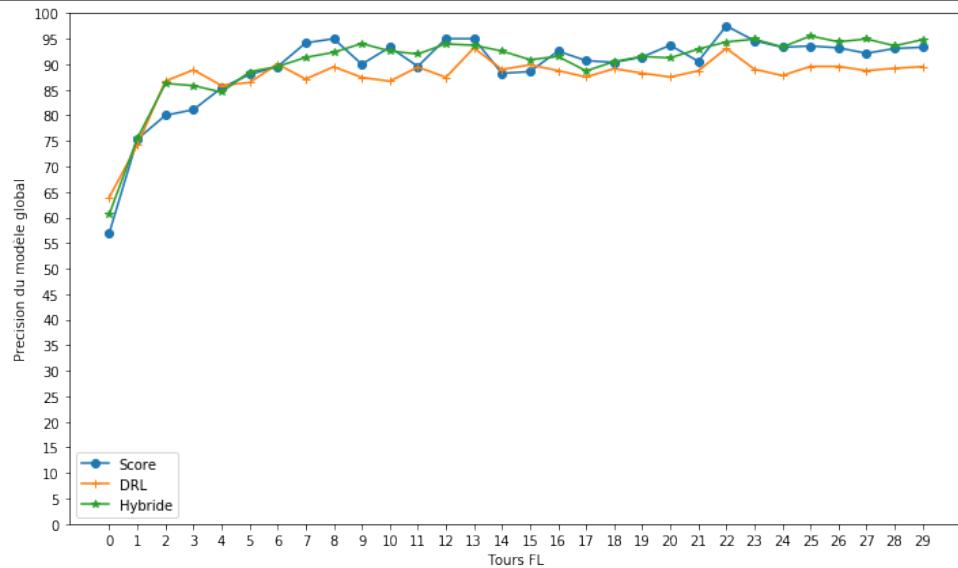


FIGURE 5.17 – Convergence du modèle global avec les trois approches dans un réseau de 300 noeuds.

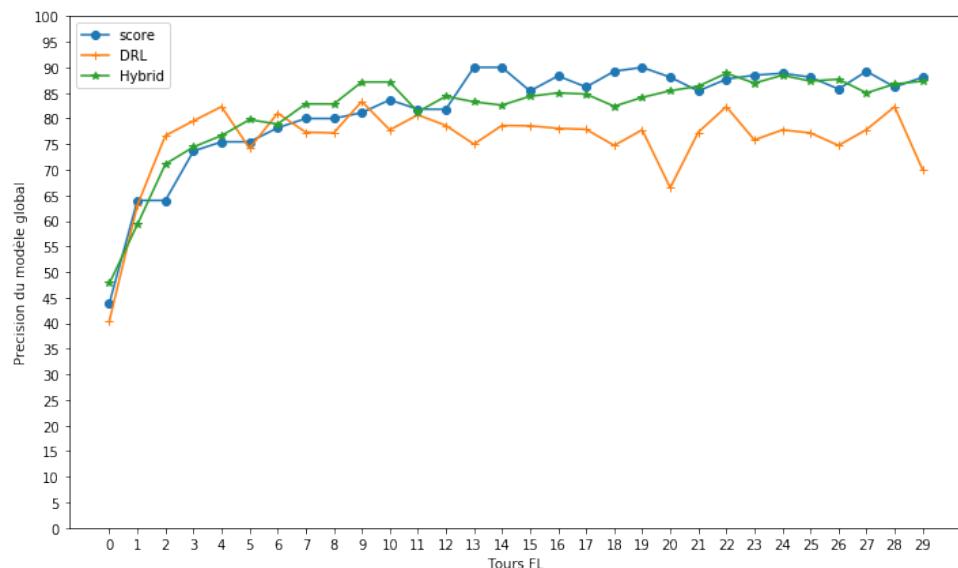


FIGURE 5.18 – Convergence du modèle global avec les trois approches dans un réseau avec un taux de noeuds malhonnêtes de 60%.

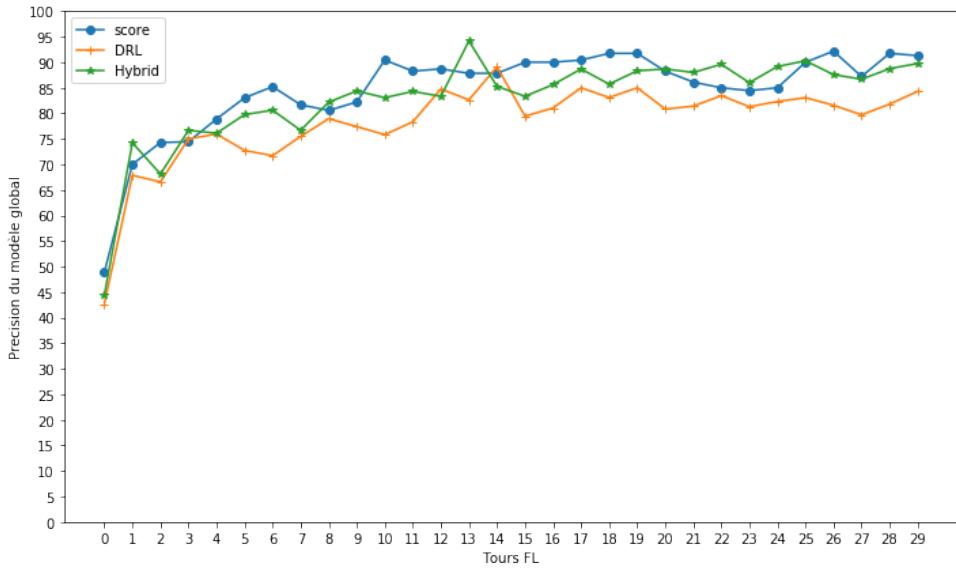


FIGURE 5.19 – Convergence du modèle global avec les trois approches dans un réseau avec un taux d'abandon de 60%.

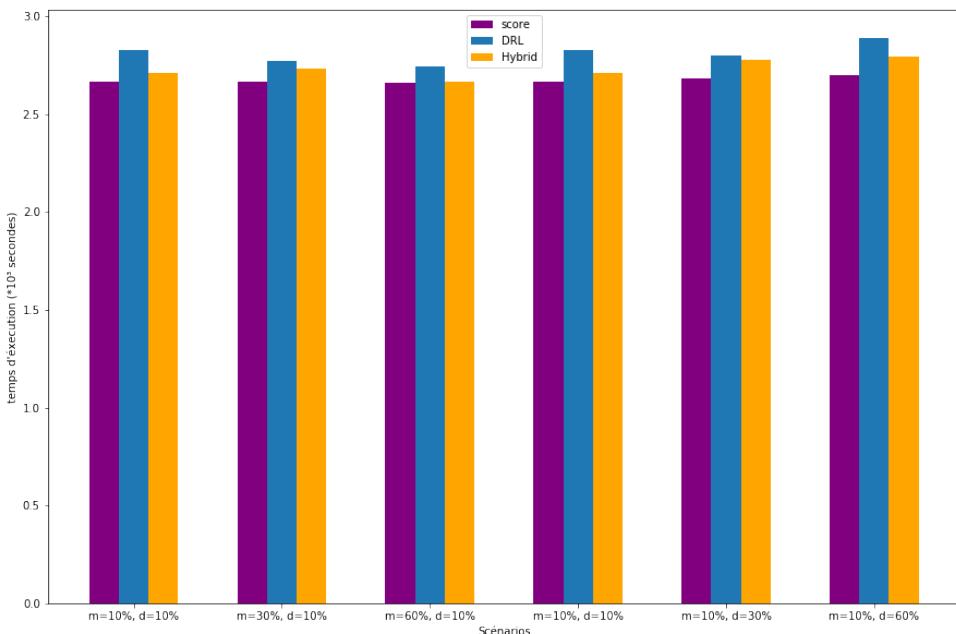


FIGURE 5.20 – Temps d'exécution des trois approches de sélection avec différents taux d'abandon et nœuds malveillants.

Les figures 5.17, 5.18 et 5.19 mettent en lumière que l'approche hybride améliore effectivement la convergence du modèle global. Par ailleurs, depuis la figure 5.20 nous remarquons que l'approche par score entraîne le plus petit temps d'exécution car elle prend en considération le coût engendré par les nœuds, ainsi que l'approche hybride, qui donne des résultats plutôt similaires grâce au pré-entraînement effectué avec les résultats

CHAPITRE 5. RÉALISATION ET ÉVALUATION

de l'approche par score.

Selon la figure 5.17 nous pouvons observer que l'approche hybride démontre une capacité de mise à l'échelle, réussissant à prédire les comportements des noeuds (abandon ou malhonnêteté) et à sélectionner les noeuds optimaux, même avec un réseau de plus grande envergure. Cette performance contraste avec les deux autres approches, qui exhibent une variation prononcée dans la convergence du modèle global, qui peut être attribuée à des sélections inadéquates des noeuds. Cette disparité s'explique par le fait que l'approche hybride, à travers son apprentissage par renforcement profond (DRL), a été entraînée de manière approfondie, lui permettant ainsi de détenir une vue globale de l'ensemble de noeuds.

Contrairement à l'approche basée sur le score, qui est fiable et rapide au début mais qui au fil du temps, tend à sélectionner invariablement le même ensemble de noeuds, altérant ainsi la nature distribuée du système, l'approche hybride offre un niveau supérieur de sophistication et d'efficacité. De plus, en raison de la simplicité de la formule sous-jacente à l'approche par score, ses résultats peuvent être aisément anticipés, offrant une opportunité aux noeuds malveillants d'exploiter cette vulnérabilité. À titre d'exemple, au sein de la figure 5.6, un noeud peut adopter un comportement favorable pendant une période déterminée afin de garantir sa sélection, avant d'engager des actions hostiles.

Depuis les figures 5.17, 5.18 et 5.19, nous observant que l'approche basée sur le renforcement profond (DRL) est moins performante que les autres mais ces résultats restent acceptables. Cela découle du mécanisme d'évaluation instauré, qui ne permet pas la contamination du modèle global par des modèles falsifiés.

Afin de garantir que les noeuds sélectionnés engendrent un minimum de coût et des modèles non falsifiés, l'approche hybride s'appuie sur la sélection par score. Le score est utilisé que pour les premières tâches d'apprentissage fédéré, en attendant que l'agent DRL acquiert une convergence adéquate pour être déployé. Cette approche garantit ainsi une performance ininterrompue dès le départ.

En outre, l'approche basée sur le score sert de moyen pour définir l'environnement offert à l'agent DRL, agissant comme un guide qui oriente l'agent vers les choix les plus optimaux, limitant ainsi le parcours de l'ensemble des cas possibles et assurant ainsi une convergence accélérée de l'agent DRL.

Par ailleurs, depuis la figure 5.19, nous constatons que, indépendamment du mécanisme de sélection adopté, la précision du modèle global ne diminue pas plus d'un certain seuil. Cette constatation est étroitement liée à l'agrégation par niveau proposée qui assure une disponibilité élevée au sein du système, et que si l'un des agrégateurs abandonne sa tâche, un autre s'en charge et ainsi les seules modèles perdus sont ceux issus des participants à l'entraînement local et non pas ceux issus des agrégateurs(modèles intermédiaires).

5.5 Conclusion

Dans ce dernier chapitre, nous avons minutieusement exposé notre mise en œuvre, en mettant en avant les outils d'implémentation, l'environnement de simulation, ainsi que les divers paramètres utilisés. Par la suite, nous avons procédé à une revue exhaustive des résultats obtenus au sein de divers scénarios. Cette démarche nous a permis d'aboutir à la conclusion selon laquelle la solution hybride présente en effet une meilleure convergence que l'approche basée sur l'apprentissage par renforcement profond (DRL), tout en offrant une adaptation plus souple et davantage de flexibilité que l'approche basée sur le score.

Conclusion Générale

L'intégration de l'apprentissage fédéré avec l'internet des objets a gagné en importance ces dernières années. L'apprentissage fédéré offre aux parties prenantes la possibilité de contribuer au processus d'apprentissage en conservant leurs données localement, garantissant ainsi la confidentialité et la vie privée. Cependant, cette approche n'est pas sans défis. Parmi les problématiques majeures, nous retrouvons la dépendance à un serveur central, créant un point de vulnérabilité unique dans le système. De plus, les enjeux de synchronicité, le risque de corruption, ainsi que les limitations en ressources des participants suscitent des inquiétudes significatives.

Pour répondre à ces vulnérabilités, plusieurs avenues prometteuses ont été explorées. En particulier, l'adoption de la blockchain émerge comme une solution robuste pour éliminer ce point de faiblesse unique et instaurer une transparence inaltérable au sein du système. Cette technologie décentralisée offre un gage de confiance et de fiabilité dans un contexte d'apprentissage fédéré. Malgré l'introduction de la blockchain, les systèmes d'apprentissage fédéré pour l'internet des objets restent vulnérables quant à la nature des participants facilement corruptibles. C'est pour cela que nous nous sommes plongés dans la question cruciale de la sélection des noeuds. Il s'est avéré que la sélection des noeuds permet d'améliorer la sécurité du système ainsi que ses performances étant donné que seuls les noeuds les plus performants et les plus honnêtes sont choisis. Dans notre état de l'art nous avons distingué deux approches principales : d'une part les approches algorithmiques, adaptées aux environnements à contraintes, exigeant peu de ressources de calcul, mais pèchent par leur rigidité face aux changements de comportement des noeuds et leur scalabilité limitée. D'autre part, les approches intelligentes se démarquent par leur adaptabilité, leur capacité à être généralisées à diverses situations et leur aptitude à gérer un grand nombre d'utilisateurs, assurant ainsi une solide scalabilité. Cependant, elles requièrent des ressources de calcul substantielles et peuvent parfois se complexifier.

Les deux familles d'approches ayant chacune ses avantages et ses inconvénients, nous avons pensé à combiner les forces des deux et ce à travers une méthode de sélection hybride. Nous avons élaboré une approche novatrice qui affronte activement les problématiques citées précédemment. Au cœur de notre démarche, nous avons conçu un système de sélection de noeuds qui se base sur un calcul de score au départ pour à la fin évoluer vers une sélection assistée par un DRL ainsi nous nous assurons d'avoir un système de sélection fonctionnel donnant des résultats satisfaisants dès son lancement en attendant de pouvoir entraîner le DRL pour plus de flexibilité et de scalabilité.

Mais notre contribution ne se résume pas seulement à ça. Nous avons mis en place un système de calcul d'honnêteté prenant en compte les comportements et les contributions

CHAPITRE 5. RÉALISATION ET ÉVALUATION

des participants. Des évaluations rigoureuses nous permettent d'identifier précisément les nœuds malveillants au sein du réseau, renforçant considérablement la sécurité de l'apprentissage fédéré dans le contexte de l'IoT.

Par ailleurs, nous avons abordé la question de la synchronicité et de la centralisation inhérente à l'apprentissage fédéré. Notre approche d'agrégation multi-niveaux offre l'asynchronicité nécessaire sans obliger les nœuds à attendre pour reprendre leurs tâches. Cette stratégie innovante décentralise également le processus d'agrégation, augmentant la disponibilité et la robustesse du système. Pour évaluer la validité de nos hypothèses, nous avons conduit une série d'expérimentations englobant une variété de scénarios distincts. Les expériences établies confirment de manière convaincante que notre approche parvient à fournir des résultats satisfaisants. Cette réussite est d'autant plus remarquable qu'elle parvient à établir un équilibre judicieux entre des principes de décentralisation, d'asynchronicité, et de scalabilité, tout en maintenant de bonnes performances dans le cadre de l'apprentissage fédéré.

Alors que ce mémoire représente une étape cruciale dans notre exploration, il est impératif de souligner les perspectives prometteuses que nos découvertes ouvrent pour l'avenir. En priorité, les mécanismes d'incitation émergent comme un enjeu essentiel. La motivation des nœuds à contribuer activement est fondamentale pour le succès de l'apprentissage fédéré. Ainsi, il serait ambitieux de poursuivre les investigations dans ce domaine pour développer des méthodes d'incitation ingénieuses qui favorisent la participation tout en garantissant la qualité des résultats.

Parallèlement, la sécurité de notre solution demeure une priorité majeure. Bien que nous ayons mis en place des mécanismes pour détecter les attaques de nœuds malveillants, des avancées en matière de détection d'anomalies et de renforcement de la sécurité peuvent être envisagées pour préserver l'intégrité de l'apprentissage fédéré dans des environnements IoT en constante mutation.

Enfin, la dimension blockchain mérite une attention approfondie. La question de la scalabilité de notre solution se pose, particulièrement à mesure que l'adoption de l'apprentissage fédéré se propage. Ainsi, l'exploration des mécanismes de mise à l'échelle offerts par la blockchain demeure cruciale pour garantir une expansion sans compromettre la sécurité ni la transparence.

Bibliographie

- ABAD, M. S. H., OZFATURA, E., GUNDUZ, D., & ERCETIN, O. (2020). Hierarchical federated learning across heterogeneous cellular networks. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8866-8870.
- ABDMEZIEM, M. R., TANDJAOUI, D., & ROMDHANI, I. (2016). Architecting the internet of things : state of the art. *Robots and Sensor Clouds*, 55-75.
- ALBRECHT, J. P. (2016). How the GDPR Will Change the World. *European Data Protection Law Review*, 2(3). <https://doi.org/10.21552/EDPL/2016/3/4>
- ALI, M., KARIMIPOUR, H., & TARIQ, M. (2021). Integration of blockchain and federated learning for Internet of Things : Recent advances and future challenges. *Computers and Security*, 108. <https://doi.org/10.1016/j.cose.2021.102355>
- AL-QASEEMI, S. A., ALMULHIM, H. A., ALMULHIM, M. F., & CHAUDHRY, S. R. (2016). IoT architecture challenges and issues : Lack of standardization, 731-738. <https://doi.org/10.1109/FTC.2016.7821686>
- ALSHEHRI, F., & MUHAMMAD, G. (2020). A comprehensive survey of the Internet of Things (IoT) and AI-based smart healthcare. *IEEE Access*, 9, 3660-3678.
- ALZUBI, J., NAYYAR, A., & KUMAR, A. (2018). Machine learning from theory to algorithms : an overview. *Journal of physics : conference series*, 1142, 012012.
- ARULKUMARAN, K., DEISENROTH, M. P., BRUNDAGE, M., & BHARATH, A. A. (2017). Deep Reinforcement Learning : A Brief Survey. *IEEE Signal Processing Magazine*, 34(6), 26-38. <https://doi.org/10.1109/MSP.2017.2743240>
- BLANCHARD, P., EL MHAMDI, E. M., GUERRAOUI, R., & STAINER, J. (2017). Machine learning with adversaries : Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.
- BOUACIDA, N., & MOHAPATRA, P. (2021). Vulnerabilities in federated learning. *IEEE Access*, 9, 63229-63249.
- CALLAWAY, E. (2003). Low power consumption features of the ieee 802.15. 4/zigbee lr-wpan standard. *Mini-tutorial, ACM Sensys*, 3, 5-7.
- CHAI, Z., ALI, A., ZAWAD, S., TRUEX, S., ANWAR, A., BARACALDO, N., ZHOU, Y., LUDWIG, H., YAN, F., & CHENG, Y. (2020). Tifl : A tier-based federated learning system. *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*, 125-136.
- CHEN, Y., & KUNZ, T. (2016). Performance evaluation of IoT protocols under a constrained wireless access network, 1-7. <https://doi.org/10.1109/MoWNet.2016.7496622>

BIBLIOGRAPHIE

- CHIK, W. B. (2013). The Singapore personal data protection act and an assessment of future trends in data privacy reform. *Computer Law and Security Review*, 29, 554-575. <https://doi.org/10.1016/j.clsr.2013.07.010>
- DABNEY, W., ROWLAND, M., BELLEMARE, M., & MUNOS, R. (2018). *Distributional reinforcement learning with quantile regression* (Nº 1).
- DAI, H.-N., ZHENG, Z., & ZHANG, Y. (2019). Blockchain for Internet of Things : A Survey. *IEEE Internet of Things Journal*, 6(5), 8076-8094. <https://doi.org/10.1109/JIOT.2019.2920987>
- DASARADHARAMI REDDY, K., & GADEKALLU, T. R. (2023). A Comprehensive Survey on Federated Learning Techniques for Healthcare Informatics (A. D. DOULAMIS, Éd.). *Computational Intelligence and Neuroscience*, 2023, 1-19. <https://doi.org/10.1155/2023/8393990>
- DINH, T. T. A., LIU, R., ZHANG, M., CHEN, G., OOI, B. C., & WANG, J. (2018). Untangling Blockchain : A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7), 1366-1385. <https://doi.org/10.1109/TKDE.2017.2781227>
- DWORK, C., MCSHERRY, F., NISSIM, K., & SMITH, A. (2006). Calibrating noise to sensitivity in private data analysis, 265-284.
- DWORK, C., & ROTH, A. (2014). The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9, 211-407.
- FAROOQ, M. U., WASEEM, M., MAZHAR, S., KHAIRI, A., & KAMAL, T. (s. d.). *A Review on Internet of Things (IoT)* (1).
- FATIMA, H., RASHEED, H., ALI, H. S., & EKRAM, H. (2020). Machine Learning in IoT Security : Current Solutions and Future Challenges. *IEEE Communications Surveys & Tutorials*, 22(3), 1686-1721. <https://doi.org/10.1109/COMST.2020.2986444>
- FERDOUS, M. S., JABED, M., CHOWDHURY, M., HOQUE, M. A., COLMAN, A. W., FERDOUS, S., HOQUE, M. A., & COLMAN, A. (s. d.). *Blockchain Consensus Algorithms : A Survey*. <https://www.researchgate.net/publication/338738073>
- FUNG, C., YOON, C. J., & BESCHASTNIKH, I. (2018). Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv :1808.04866*.
- GAUR, A., SCOTNEY, B., PARR, G., & McCLEAN, S. (2015). Smart city architecture and its applications based on IoT. *Procedia computer science*, 52, 1089-1094.
- GÉRON, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc."
- GHOSH, A., CHUNG, J., YIN, D., & RAMCHANDRAN, K. (2022). An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12), 8076-8091.
- GILAD, Y., HEMO, R., MICALI, S., VLACHOS, G., & ZELDOVICH, N. (2017). Algorand : Scaling byzantine agreements for cryptocurrencies, 51-68.
- GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. (2016). *Deep learning*. MIT press.
- GU, J., SUN, B., DU, X., WANG, J., ZHUANG, Y., & WANG, Z. (2018). Consortium Blockchain-Based Malware Detection in Mobile Devices. *IEEE Access*, 6, 12118-12128. <https://doi.org/10.1109/ACCESS.2018.2805783>

BIBLIOGRAPHIE

-
- HABIB, G., SHARMA, S., IBRAHIM, S., AHMAD, I., QURESHI, S., & ISHFAQ, M. (2022). Blockchain Technology : Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. *Future Internet*, 14(11). <https://doi.org/10.3390/fi14110341>
- HASSAN, M. U., REHMANI, M. H., & CHEN, J. (2019). Privacy preservation in blockchain based IoT systems : Integration issues, prospects, challenges, and future research directions. *Future Generation Computer Systems*, 97, 512-529. <https://doi.org/10.1016/j.future.2019.02.060>
- HATAMIZADEH, A., YIN, H., MOLCHANOV, P., MYRONENKO, A., LI, W., DOGRA, P., FENG, A., FLORES, M. G., KAUTZ, J., XU, D., & ROTH, H. R. (2021). Towards Understanding the Risks of Gradient Inversion in Federated Learning.
- HERMANN, M., PENTEK, T., OTTO, B., et al. (2015). Design principles for Industrie 4.0 scenarios : a literature review. *Technische Universität Dortmund, Dortmund*, 45.
- HOU, Z., CHEN, H., LI, Y., & VUCETIC, B. (2017). Incentive mechanism design for wireless energy harvesting-based Internet of Things. *IEEE Internet of Things Journal*, 5(4), 2620-2632.
- HUANG, H., KONG, W., ZHOU, S., ZHENG, Z., & GUO, S. (2021). A Survey of State-of-The-Art on Blockchains. *ACM Computing Surveys*, 54. <https://doi.org/10.1145/3441692>
- ISSA, W., MOUSTAFA, N., TURNBULL, B., SOHRABI, N., & TARI, Z. (2023). Blockchain-Based Federated Learning for Securing Internet of Things : A Comprehensive Survey. *ACM Comput. Surv.*, 55(9). <https://doi.org/10.1145/3560816>
- JAGANNATH, J., POLOSKY, N., JAGANNATH, A., RESTUCCIA, F., & MELODIA, T. (2019). Machine Learning for Wireless Communications in the Internet of Things : A Comprehensive Survey. <https://doi.org/10.1016/j.adhoc.2019.101913>
- KAELBLING, L. P., LITTMAN, M. L., & MOORE, A. W. (1996). Reinforcement learning : A survey. *Journal of artificial intelligence research*, 4, 237-285.
- KANG, J., XIONG, Z., NIYATO, D., XIE, S., & ZHANG, J. (2019). Incentive Mechanism for Reliable Federated Learning : A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet of Things Journal*, 6, 10700-10714. <https://doi.org/10.1109/JIOT.2019.2940820>
- KANKANHALLI, A., CHARALABIDIS, Y., & MELLOULI, S. (2019). IoT and AI for smart government : A research agenda. *Government Information Quarterly*, 36(2), 304-309.
- KHAN, A., THIJ, M. t., & WILBIK, A. (2022). Vertical Federated Learning : A Structured Literature Review. *arXiv preprint arXiv:2212.00622*.
- KHAN, L. U., SAAD, W., HAN, Z., HOSSAIN, E., & HONG, C. S. (2021). Federated learning for internet of things : Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys and Tutorials*, 23, 1759-1799. <https://doi.org/10.1109/COMST.2021.3090430>
- KHAN, M., den HARTOG, F., & HU, J. (2022). A Survey and Ontology of Blockchain Consensus Algorithms for Resource-Constrained IoT Systems. *Sensors*, 22. <https://doi.org/10.3390/s22218188>

BIBLIOGRAPHIE

- KOLB, J., ABDELBAKY, M., KATZ, R. H., & CULLER, D. E. (2020). Core concepts, challenges, and future directions in blockchain : A centralized tutorial. *ACM Computing Surveys*, 53. <https://doi.org/10.1145/3366370>
- LEE, I. (2017). An exploratory study of the impact of the internet of things (IoT) on business model innovation : Building smart enterprises at fortune 500 companies. In *The Internet of Things : Breakthroughs in Research and Practice* (p. 423-440). IGI Global.
- LI, Q., WEN, Z., WU, Z., HU, S., WANG, N., LI, Y., LIU, X., & HE, B. (2021). A Survey on Federated Learning Systems : Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering*. <https://doi.org/10.1109/TKDE.2021.3124599>
- LI, S., XU, L. D., & ZHAO, S. (2015). The internet of things : a survey. *Information Systems Frontiers*, 17, 243-259. <https://doi.org/10.1007/s10796-014-9492-7>
- LI, Z., HE, Y., YU, H., KANG, J., LI, X., XU, Z., & NIYATO, D. (2022). Data heterogeneity-robust federated learning via group client selection in industrial iot. *IEEE Internet of Things Journal*, 9(18), 17844-17857.
- LIANG, Y.-C. (2020, janvier). *Blockchain for Dynamic Spectrum Management* [accessed 6 Nov, 2022]. https://www.researchgate.net/figure/The-structure-of-a-Blockchain-A-block-is-composed-of-a-header-and-a-body-where-a-header_fig1_337306138
- LIU, Y., LI, K., JIN, Y., ZHANG, Y., & QU, W. (2011). A novel reputation computation model based on subjective logic for mobile ad hoc networks. *Future Generation Computer Systems*, 27(5), 547-554.
- LU, Y., HUANG, X., ZHANG, K., MAHARJAN, S., & ZHANG, Y. (2020). Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Transactions on Vehicular Technology*, 69(4), 4298-4311.
- LU, Y., HUANG, X., ZHANG, K., MAHARJAN, S., & ZHANG, Y. (2021). Blockchain and Federated Learning for 5G beyond. *IEEE Network*, 35, 219-225. <https://doi.org/10.1109/MNET.011.1900598>
- LUDWIG, H., & BARACALDO, N. (2022). *Federated Learning : A Comprehensive Overview of Methods and Applications*. Springer.
- MA, C., LI, J., SHI, L., DING, M., WANG, T., HAN, Z., & POOR, H. V. (2022). When federated learning meets blockchain : A new distributed learning paradigm. *IEEE Computational Intelligence Magazine*, 17(3), 26-33.
- MA, J., NAAS, S.-A., SIGG, S., & LYU, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*, 37(9), 5880-5901. <https://doi.org/https://doi.org/10.1002/int.22818>
- MAHDAVINEJAD, M. S., REZVAN, M., BAREKATAIN, M., ADIBI, P., BARNAGHI, P., & SHETH, A. P. (2018). Machine learning for internet of things data analysis : a survey. *Digital Communications and Networks*, 4(3), 161-175. <https://doi.org/https://doi.org/10.1016/j.dcan.2017.10.002>
- MAHESH, B. (2018). Machine Learning Algorithms-A Review. *International Journal of Science and Research*. <https://doi.org/10.21275/ART20203995>
- MAHESH, B. (2019). Machine Learning Algorithms -A Review. <https://doi.org/10.21275/ART20203995>

BIBLIOGRAPHIE

- MELIS, L., SONG, C., DE CRISTOFARO, E., & SHMATIKOV, V. (2019). Exploiting Unintended Feature Leakage in Collaborative Learning, 691-706. <https://doi.org/10.1109/SP.2019.00029>
- MILES, B., BOURENNANE, E. B., BOUCHERKHA, S., & CHIKHI, S. (2020). A study of LoRaWAN protocol performance for IoT applications in smart agriculture. *Computer Communications*, 164, 148-157. <https://doi.org/10.1016/j.comcom.2020.10.009>
- MISRA, N., DIXIT, Y., AL-MALLAHI, A., BHULLAR, M. S., UPADHYAY, R., & MARTYNENKO, A. (2020). IoT, big data and artificial intelligence in agriculture and food industry. *IEEE Internet of Things Journal*.
- MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A. K., OSTROVSKI, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
- MOHAMMED, I., TABATABAI, S., AL-FUQAH, A., EL BOUANANI, F., QADIR, J., QOLOMANY, B., & GUIZANI, M. (2020). Budgeted online selection of candidate IoT clients to participate in federated learning. *IEEE Internet of Things Journal*, 8(7), 5938-5952.
- MORI, J., TERANISHI, I., & FURUKAWA, R. (2022). Continual Horizontal Federated Learning for Heterogeneous Data. *2022 International Joint Conference on Neural Networks (IJCNN)*. <https://doi.org/10.1109/ijcnn55064.2022.9892815>
- MOTHUKURI, V., PARIZI, R. M., POURIYEH, S., HUANG, Y., DEHGHANTANHA, A., & SRIVASTAVA, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115, 619-640.
- NAKAMOTO, S. (2008). *Bitcoin : A Peer-to-Peer Electronic Cash System*. www.bitcoin.org
- NG, A. Y., & JORDAN, M. I. (2013). PEGASUS : A policy search method for large MDPs and POMDPs. *arXiv preprint arXiv :1301.3878*.
- NGUYEN, D. C., DING, M., PATHIRANA, P. N., SENEVIRATNE, A., LI, J., & POOR, H. V. (2021). Federated Learning for Internet of Things : A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*, 23, 1622-1658. <https://doi.org/10.1109/COMST.2021.3075439>
- NIKNAM, S., DHILLON, H. S., & REED, J. H. (2020). Federated learning for wireless communications : Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6), 46-51.
- NISHIO, T., & YONETANI, R. (2018). Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. <https://doi.org/10.1109/ICC.2019.8761315>
- NOFER, M., GOMBER, P., HINZ, O., & SCHIERECK, D. (2017). Blockchain. *Business and Information Systems Engineering*, 59, 183-187. <https://doi.org/10.1007/s12599-017-0467-3>
- OTOUM, S., RIDHAWI, I. A., & MOUFTAH, H. (2022). Securing Critical IoT Infrastructures with Blockchain-Supported Federated Learning. *IEEE Internet of Things Journal*, 9, 2592-2601. <https://doi.org/10.1109/JIOT.2021.3088056>
- PARK, J., & LIM, H.-K. (2022). Privacy-Preserving Federated Learning Using Homomorphic Encryption. *Applied Sciences*. <https://api.semanticscholar.org/CorpusID:245922630>

BIBLIOGRAPHIE

- PATEL, K. K., PATEL, S. M., & SCHOLAR, P. G. (2016). *Internet of Things-IOT : Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*. <http://ijesc.org/>
- PUTHAL, D., MALIK, N., MOHANTY, S. P., KOUGIANOS, E., & DAS, G. (2018). Everything You Wanted to Know About the Blockchain : Its Promise, Components, Processes, and Problems. *IEEE Consumer Electronics Magazine*, 7(4), 6-14. <https://doi.org/10.1109/MCE.2018.2816299>
- QIU, J., WU, Q., DING, G., XU, Y., & FENG, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1), 67.
- QU, Z., GUO, S., WANG, H., YE, B., WANG, Y., ZOMAYA, A. Y., & TANG, B. (2022). Partial Synchronization to Accelerate Federated Learning Over Relay-Assisted Edge Networks. *IEEE Transactions on Mobile Computing*, 21(12), 4502-4516. <https://doi.org/10.1109/TMC.2021.3083154>
- RAMU, S. P., BOOPALAN, P., PHAM, Q.-V., MADDIKUNTA, P. K. R., HUYNH-THE, T., ALAZAB, M., NGUYEN, T. T., & GADEKALLU, T. R. (2022). Federated learning enabled digital twins for smart cities : Concepts, recent advances, and future directions. *Sustainable Cities and Society*, 79, 103663. <https://doi.org/https://doi.org/10.1016/j.scs.2021.103663>
- RAO, D., CHANDRA, D., & KUMAR, D. (2017). A Survey on Machine Learning : Concept, Algorithms and Applications. *International Conference on Innovation Research in Computer and Communication Engineering*.
- RAVISHANKAR, N. R., & VIJAYAKUMAR, M. V. (2017). Reinforcement Learning Algorithms : Survey and Classification. *Indian Journal of Science and Technology*, 10. <https://doi.org/10.17485/ijst/2017/v10i1/109385>
- RJOUB, G., WAHAB, O. A., BENTAHAR, J., & BATAINEH, A. (2022). Trust-driven reinforcement selection strategy for federated learning on IoT devices. *Computing*. <https://doi.org/10.1007/s00607-022-01078-1>
- RUDER, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.
- SALIMITARI, M., & CHATTERJEE, M. (2018). A Survey on Consensus Protocols in Blockchain for IoT Networks. <http://arxiv.org/abs/1809.05613>
- SAMIZADEH NIKOUI, T., RAHMANI, A. M., BALADOR, A., & HAJ SEYYED JAVADI, H. (2021). Internet of Things architecture challenges : A systematic review [e4678 IJCS-19-1067.R1]. *International Journal of Communication Systems*, 34 (4), e4678. <https://doi.org/https://doi.org/10.1002/dac.4678>
- SAMUEL, A. L. (1988). Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. In D. N. L. LEVY (Éd.), *Computer Games I* (p. 366-400). Springer New York. https://doi.org/10.1007/978-1-4613-8716-9_15
- SHARMA, P. K., GOPE, P., & PUTHAL, D. (2022). Blockchain and Federated Learning-enabled Distributed Secure and Privacy-preserving Computing Architecture for IoT Network. *Proceedings - 7th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2022*, 1-9. <https://doi.org/10.1109/EuroSPW55150.2022.00008>

BIBLIOGRAPHIE

- SHAYAN, M., FUNG, C., YOON, C. J., & BESCHASTNIKH, I. (2018). Biscotti : A ledger for private and secure peer-to-peer machine learning. *arXiv preprint arXiv :1811.09904*.
- SUO, H., WAN, J., ZOU, C., & LIU, J. (2012). Security in the Internet of Things : A Review. *3*, 648-651. <https://doi.org/10.1109/ICCSEE.2012.373>
- SVIRIDENKO, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, *32*(1), 41-43.
- THOMAS, S. F. (1989). Who solved the secretary problem ? *Statistical Science*, *4*(3), 282.
- TOURNIER, J., LESUEUR, F., MOUËL, F. L., GUYON, L., & BEN-HASSINE, H. (2021). A survey of IoT protocols and their security issues through the lens of a generic IoT stack. *Internet of Things (Netherlands)*, *16*. <https://doi.org/10.1016/j.iot.2020.100264>
- UL HASSAN, M., REHMANI, M. H., & CHEN, J. (2018). Differential Privacy Techniques for Cyber Physical Systems : A Survey.
- VAN HASSELT, H., GUEZ, A., & SILVER, D. (2016). Deep reinforcement learning with double q-learning. *Proceedings of the AAAI conference on artificial intelligence*, *30*(1).
- van ENGELEN, J. E., & HOOS, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373-440.
- VIRIYASITAVAT, W., & HOONSOPON, D. (2019a). Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration*, *13*, 32-39. <https://doi.org/10.1016/j.jii.2018.07.004>
- VIRIYASITAVAT, W., & HOONSOPON, D. (2019b). Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration*, *13*, 32-39. <https://doi.org/https://doi.org/10.1016/j.jii.2018.07.004>
- WANG, C., CHEN, J., YANG, Y., MA, X., & LIU, J. (2022). Poisoning attacks and countermeasures in intelligent networks : Status quo and prospects. *Digital Communications and Networks*, *8*(2), 225-234. <https://doi.org/https://doi.org/10.1016/j.dcan.2021.07.009>
- WANG, H., YUROCHKIN, M., SUN, Y., PAPALIOPOULOS, D., & KHAZAENI, Y. (2020). Federated learning with matched averaging. *arXiv preprint arXiv :2002.06440*.
- WANG, J., GUO, S., XIE, X., & QI, H. (2022). Protect Privacy from Gradient Leakage Attack in Federated Learning. *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 580-589. <https://doi.org/10.1109/INFOCOM48880.2022.9796841>
- WANG, W., HOANG, D. T., HU, P., XIONG, Z., NIYATO, D., WANG, P., WEN, Y., & KIM, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access*, *7*, 22328-22370.
- WEI, K., LI, J., DING, M., MA, C., YANG, H. H., FAROKHI, F., JIN, S., QUEK, T. Q., & POOR, H. V. (2020). Federated learning with differential privacy : Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, *15*, 3454-3469.
- WEN, J., ZHANG, Z., LAN, Y., CUI, Z., CAI, J., & ZHANG, W. (2022). A survey on federated learning : challenges and applications. *International Journal of Machine Learning and Cybernetics*, 1-23.

BIBLIOGRAPHIE

-
- WIRTH, C., NEUMANN, G., & FÜRNKRANZ, J. (2017). *A Survey of Preference-Based Reinforcement Learning Methods*. <http://jmlr.org/papers/v18/16-634.html>.
- XIAO, Y., ZHANG, N., LOU, W., & HOU, Y. T. (2020). A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Communications Surveys & Tutorials*, 22(2), 1432-1465. <https://doi.org/10.1109/COMST.2020.2969706>
- XU, X., WEBER, I., & STAPLES, M. (2019). *Architecture for Blockchain Applications*. Springer Cham. <https://doi.org/https://doi.org/10.1007/978-3-030-03035-3>
- YEOW, K., GANI, A., AHMAD, R. W., RODRIGUES, J. J. P. C., & KO, K. (2018). Decentralized Consensus for Edge-Centric Internet of Things : A Review, Taxonomy, and Research Issues. *IEEE Access*, 6, 1513-1524. <https://doi.org/10.1109/ACCESS.2017.2779263>
- ZHANG, C., XIE, Y., BAI, H., YU, B., LI, W., & GAO, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, 216. <https://doi.org/10.1016/j.knosys.2021.106775>
- ZHANG, H., XIE, Z., ZAREI, R., WU, T., & CHEN, K. (2021). Adaptive Client Selection in Resource Constrained Federated Learning Systems : A Deep Reinforcement Learning Approach. *IEEE Access*, 9, 98423-98432. <https://doi.org/10.1109/ACCESS.2021.3095915>
- ZHANG, J., WU, Y., & PAN, R. (2021). Incentive mechanism for horizontal federated learning based on reputation and reverse auction. *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, 947-956. <https://doi.org/10.1145/3442381.3449888>
- ZHANG, K., SONG, X., ZHANG, C., & YU, S. (2022). Challenges and future directions of secure federated learning : a survey. *Frontiers of computer science*, 16, 1-8.
- ZHANG, Y., & VAN DER SCHAAR, M. (2012). Reputation-based incentive protocols in crowdsourcing applications. *2012 Proceedings IEEE INFOCOM*, 2140-2148.
- ZHAO, Y., ZHAO, J., JIANG, L., TAN, R., NIYATO, D., LI, Z., LYU, L., & LIU, Y. (2020). Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet of Things Journal*, 8(3), 1817-1829.

Webographie

- CASTAGNO, L. (2022). Gradient descent in unsupervised learning [Accessed : Feb. 22, 2023]. <https://medium.com/@lorenzojcdcv/gradient-descent-in-unsupervised-learning-aa399f9c5f64>
- DEJESUS, T. (2022). What Is Tokenization and How Does It Work ? [Accessed : Feb. 19, 2023]. <https://www.nasdaq.com/articles/what-is-tokenization-and-how-does-it-work>
- Introduction to Machine Learning [Accessed : Feb. 22, 2023]. (2022). <https://developers.google.com/machine-learning/intro-to-ml/>
- JAIN, N. (2019). An overview of the Gradient Descent algorithm [Accessed : Feb. 21, 2023]. <https://towardsdatascience.com/an-overview-of-the-gradient-descent-algorithm-8645c9e4de1e>
- LIU, C. (2022). 5 Concepts You Should Know About Gradient Descent and Cost Function [Accessed : Feb. 22, 2023]. <https://www.kdnuggets.com/2020/05/5-concepts-gradient-descent-cost-function.html>
- MACTAGGART, A., & MACTAGGART, C. (2021, février). California Privacy Rights Act : Californians for consumer privacy. <https://www.caprivity.org/>
- MAHESHKAR, S. (2022). What Is Cross Entropy Loss ? A Tutorial With Code [Accessed : Feb. 20, 2023]. <https://wandb.ai/sauravmaheshkar/cross-entropy/reports/What-Is-Cross-Entropy-Loss-A-Tutorial-With-Code--VmlldzoxMDA5NTMx>
- MAYO, M. (2022). Frameworks for Approaching the Machine Learning Process [Accessed : Feb. 22, 2023]. <https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html>
- PARMAR, R. (2018). Common Loss functions in machine learning [Accessed : Feb. 20, 2023]. <https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23>
- STAFF, C. (2021). What Is Tokenization in Blockchain ? [Accessed : Feb. 19, 2023]. <https://www.gemini.com/en-US/cryptopedia/what-is-tokenization-definition-crypto-token>

Annexe

A Concepts liés à l'apprentissage automatique

A.1 Les fonctions de pertes

L'apprentissage automatique utilise un ensemble de données d'apprentissage qui comprend des entrées et des sorties correctes, afin de permettre au modèle d'apprendre au fil du temps. Donc les algorithmes (supervisé et non supervisé) de ML mesurent leur performance à l'aide de la fonction de perte et s'ajuste avec des méthodes d'optimisation jusqu'à ce que l'erreur soit suffisamment minimisée. Cependant, certains algorithmes d'apprentissage non supervisé tel que le clustering (CASTAGNO, 2022), et l'apprentissage par renforcement n'utilisent pas une fonction de coût pour mesurer la performance. Dans ce qui suit nous allons voir les fonctions de pertes les plus utilisées pour chaque situation (la classification et la régression) (PARMAR, 2018), :

Fonctions de perte dans le cas d'une classification :Multi class SVM Loss (Hinge Loss) est utilisée pour les classifications à marge maximale et les SVM (machines à vecteur support) tel qu'illustré dans la figure 21. Voici sa formule mathématique :

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta) \quad (1)$$

L_i est la perte pour le i-ème exemple. s_j est le score attribué à la j-ème classe pour le i-ème exemple. y_i est la véritable étiquette de classe pour le i-ème exemple. Δ est un hyperparamètre de marge, qui détermine la différence minimale de score requise entre la classe correcte et les autres classes pour que le modèle soit pénalisé. L'objectif de la fonction de perte SVM multi-classes est de s'assurer que les scores attribués à la classe correcte sont supérieurs aux scores attribués à toutes les autres classes d'au moins Δ . Si ce n'est pas le cas, le modèle est pénalisé, la pénalité étant proportionnelle à la mesure dans laquelle les scores violent cette contrainte. **Cross-Entropy Loss** est une fonction de perte fréquemment utilisée en apprentissage automatique (MAHESHKAR, 2022), elle est définie comme suit :

(b) — dans le cas d'une classification binaire :

$$L(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \quad (2)$$

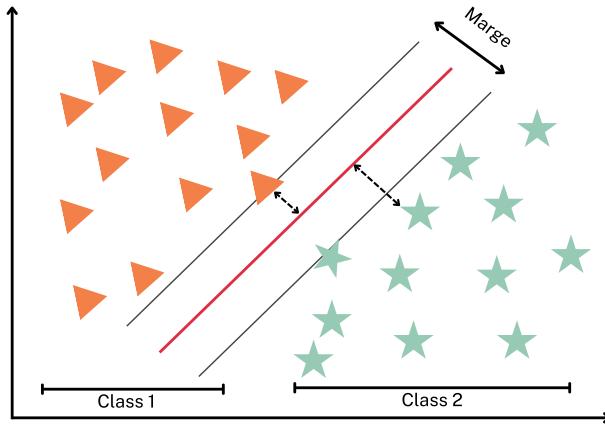


FIGURE 21 – Exemple de machine à vecteur support

— dans le cas de N Classes :

$$L(y, \hat{y}) = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (3)$$

y est la véritable étiquette ou distribution cible, représentée sous la forme d'un vecteur à un cout (un vecteur de zéros avec un seul 1 indiquant la véritable classe). p est la probabilité prédictive. \hat{y} est la distribution de probabilité prédictive sur les classes, généralement obtenue en faisant passer l'entrée par une fonction softmax pour normaliser les scores en probabilités. n est le nombre de classes. L'intuition derrière la perte d'entropie croisée est qu'elle mesure la différence entre la distribution réelle et la distribution prédictive. En minimisant cette différence, le modèle est encouragé à attribuer des probabilités élevées à la classe correcte et des probabilités faibles aux autres classes.

1. Fonctions de perte dans le cas d'une régression :

- (a) **Mean Square Error (Quadratic Loss)** mesure la différence moyenne au carré entre les valeurs prédictives et réelles d'une variable continue. La formule de MSE est la suivante :

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4)$$

N est le nombre d'exemples dans l'ensemble de données. y_i est la valeur réelle de la variable cible pour le i -ème exemple. \hat{y}_i est la valeur prédictive de la variable cible pour le i -ème exemple. La finalité de l'erreur quadratique moyenne est de minimiser la différence quadratique moyenne entre les valeurs prédictives et réelles de la variable cible. En minimisant la MSE, nous trouvons effectivement les paramètres du modèle qui s'adaptent le mieux aux données d'apprentissage

dans le sens des moindres carrés. La MSE est couramment utilisée dans la régression linéaire et d'autres problèmes de régression.

- (b) **Mean Absolute Error (L1 Loss)** est une autre fonction de perte largement utilisée dans les problèmes de régression. Elle mesure la différence absolue moyenne entre les valeurs prédites et réelles d'une variable continue. La formule de la MAE est la suivante :

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (5)$$

N est le nombre d'exemples dans l'ensemble de données. y_i est la valeur réelle de la variable cible pour le i -ème exemple. \hat{y}_i est la valeur prédite de la variable cible pour le i -ème exemple. Le but de l'erreur absolue moyenne est de minimiser la différence absolue moyenne entre les valeurs prédites et réelles de la variable cible. La MAE est couramment utilisée dans la régression robuste et d'autres problèmes de régression où la présence de valeurs aberrantes ou d'erreurs importantes est un problème.

Pour minimiser ces fonctions de perte, les algorithmes d'optimisation itératifs sont largement utilisés, tels que les algorithmes de descente du gradient : Batch gradient descent, Stochastic gradient descent, Adagrad, Momentum gradient descent et Mini-batch gradient descent (RUDER, 2016).

A.2 La descente du gradient

La descente du gradient fonctionne en mettant à jour de manière itérative les paramètres du modèle dans la direction du gradient négatif de la fonction de perte. En déplaçant les paramètres dans la direction du gradient négatif, nous pouvons progressivement la valeur de la fonction de perte et trouver les paramètres optimaux qui la minimisent

(JAIN, 2019). Une fois que nous avons le gradient, nous pouvons mettre à jour les paramètres du modèle comme suit (C. LIU, 2022) :

$$\theta_{i+1} = \theta_i - \eta \nabla_{\theta} L(\theta_i) \quad (6)$$

θ_i est le vecteur des paramètres du modèle à l'itération i . $\nabla_{\theta} L(\theta_i)$ est le gradient de la fonction de perte par rapport aux paramètres du modèle à l'itération i . η est le taux d'apprentissage, qui contrôle la taille du pas de la mise à jour.

A.3 Le taux d'apprentissage

Le taux d'apprentissage détermine la taille du pas que nous faisons dans la direction du gradient négatif à chaque itération. Si le taux d'apprentissage est trop petit, l'algorithme peut prendre beaucoup de temps pour converger.

rithme peut converger très lentement, tandis que s'il est trop grand, l'algorithme peut osciller ou même diverger, tel qu'illustré par la figure 22.

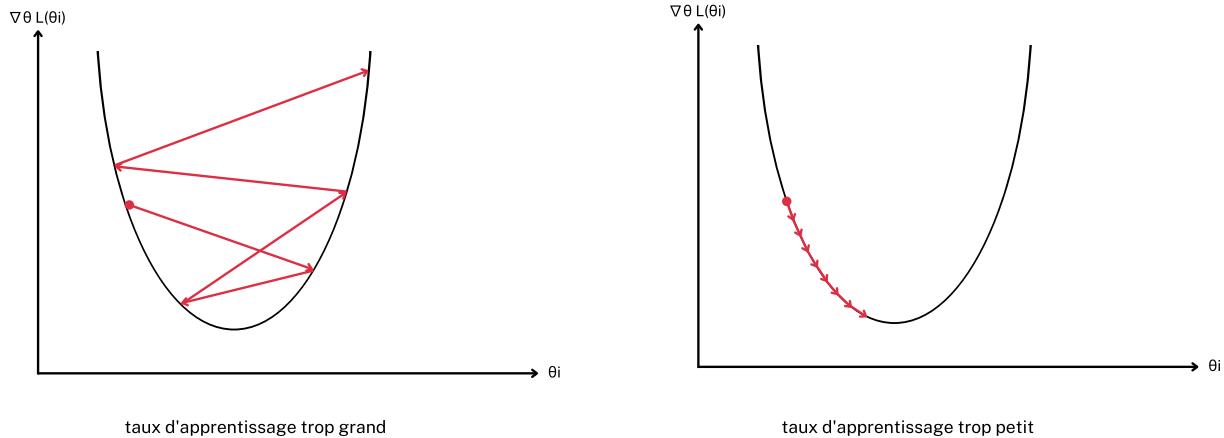


FIGURE 22 – L'effet du taux d'apprentissage sur la descente du gradient

A.4 Autres catégories de ML

Apprentissage semi-supervisé

Dans les deux types supervisé et non supervisé, soit il n'y a pas d'étiquettes pour toutes les observations dans l'ensemble de données, soit des étiquettes sont présentes pour toutes les observations. L'apprentissage semi-supervisé se situe entre ces deux types. Dans de nombreuses situations pratiques, le coût de l'étiquetage est assez élevé, car il faut des experts humains qualifiés pour le faire. Ainsi, en l'absence d'étiquettes dans la majorité des observations mais présentes dans quelques unes, les algorithmes semi-supervisés sont les meilleurs candidats pour la construction de modèles (van ENGELEN & HOOS, 2020).

Réseau de neurones

Les réseaux neuronaux sont un type de modèle d'apprentissage automatique qui s'inspire de la structure et du fonctionnement du cerveau humain. Ils se composent de noeuds interconnectés, appelés neurones, organisés en couches. Chaque neurone reçoit des données d'autres neurones ou de sources externes et applique une fonction mathématique à ces données pour générer une sortie. La sortie de chaque neurone est ensuite transmise aux neurones de la couche suivante. Les connexions entre les neurones sont représentées par des poids, qui peuvent être ajustés pendant l'entraînement à l'aide d'un processus appelé rétro-propagation (backpropagation) (FATIMA et al., 2020). La rétropropagation consiste à calculer l'erreur entre la sortie du réseau et la sortie souhaitée, puis à utiliser cette erreur pour ajuster les poids de manière à la minimiser.

Il existe de nombreux types de réseaux neuronaux, chacun adapté à différents types de problèmes. Voici quelques exemples (GÉRON, 2022 ; GOODFELLOW et al., 2016) :

- **Feedforward neural networks** : c'est le type de réseau neuronal le plus simple, dans lequel les informations circulent dans une seule direction, de l'entrée à la sortie. Ils sont souvent utilisés pour des tâches telles que la classification ou la régression.
- **Convolutional neural networks (CNN)** : Ils sont conçus pour les tâches de reconnaissance d'images, où l'entrée est une image bidimensionnelle. Ils utilisent une série de couches convolutionnelles pour extraire les caractéristiques de l'image, puis une ou plusieurs couches entièrement connectées pour effectuer une prédiction.
- **Recurrent neural networks (RNN)** : Ils sont conçus pour les tâches qui impliquent des données séquentielles, comme les séries chronologiques ou le traitement du langage naturel. Ils comportent des boucles qui permettent de faire passer les informations d'un pas de temps à l'autre, ce qui leur permet de saisir les dépendances temporelles.
- **Generative adversarial networks (GAN)** : Il s'agit d'un type de réseau neuronal utilisé pour des tâches de génération, comme la génération d'images ou de textes. Ils se composent de deux réseaux : un réseau génératrice qui génère des échantillons, et un réseau discriminateur qui tente de distinguer les échantillons réels des faux. Les deux réseaux sont formés ensemble dans un processus appelé formation contradictoire.

Apprentissage ensembliste

Le but principal d'une méthode ensembliste est l'intégration de plusieurs modèles faibles qui sont formés séparément afin de renforcer et d'améliorer la généralisation et la robustesse par rapport à un modèle unique (RAO et al., 2017). D'autres applications de ce type d'apprentissage incluent l'attribution d'une confiance à la décision prise par le modèle, la sélection des caractéristiques optimaux, la fusion de données, l'apprentissage incrémental, l'apprentissage non stationnaire et la correction d'erreurs (MAHESH, 2018).

Apprentissage multi-tâches

L'objectif de l'apprentissage multi-tâche est de résoudre multiples tâches différentes en même temps, en tirant parti des similitudes entre les différentes tâches. Cela peut améliorer l'efficacité de l'apprentissage et également servir à régulariser. Cet apprentissage est effectif dans le cas de N tâches qui sont corrélées entre elles mais pas exactement identiques, car il permet d'améliorer l'apprentissage d'un modèle particulier en utilisant les connaissances contenues dans l'ensemble de ces N tâches (MAHESH, 2018).

Apprentissage basé sur les instances

L'apprentissage basé sur les instances désigne une famille de techniques de classification et de régression, qui permettent de classifier ou prédire en se basant sur la similarité entre la requête et les instances les plus proche(s) dans l'ensemble d'apprentissage. Cette technique stocke toutes les données, et au moment d'une requête, elle déduit une réponse à partir d'un examen des voisins les plus proches de la requête. Un exemple de cette

méthode, l'algorithme des K plus proches voisins (KNN), c'est un algorithme d'apprentissage supervisé, qui peut être utilisé pour une classification comme pour une régression (MAHESH, 2018).

B Concepts liés à la blockchain

B.1 Mineurs

Les mineurs d'une blockchain sont les participants qui dédient leurs ressources pour vérifier les transactions et ensuite les rajouter à la blockchain. Leur travail (Mining) est essentiel à la blockchain car il permet de maintenir la sécurité et l'intégrité du réseau. Dans une blockchain publique, tous les participants peuvent devenir des mineurs, cependant dans une blockchain privée l'admission de nouveaux mineurs est contrôlée par les autorités dirigeantes de la blockchain (XU et al., 2019).

B.2 Consensus

Le consensus est le processus par lequel les participants d'un réseau blockchain s'accordent sur l'état du registre. Dans un réseau décentralisé de blockchain, le consensus est obtenu grâce à un algorithme de consensus qui garantit que tous les nœuds du réseau se mettent d'accord sur l'ordre et la validité des transactions. C'est un élément essentiel d'un réseau blockchain car il garantit que le registre est infalsifiable et résistant aux attaques (VIRIYASITAVAT & HOONSOPON, 2019b). De multiples algorithmes de consensus ont été proposés, parmi eux : Proof of Work, Proof of Stake et Byzantine Fault Tolerance.

B.3 Contrat intelligent

Les contrats intelligents sont des programmes déployés sur la blockchain comme des données, et qui peuvent être exécutés durant les transactions. Leur code est déterministe et non modifiable une fois déployé. Ils peuvent garder ou transférer des actifs numériques comme ils peuvent invoquer d'autres contrats stockés sur la blockchain. Ils sont utilisés pour administrer la propriété des actifs. Bien que les contrats intelligents ne soient pas toujours utilisés pour établir des contrats légaux, ils peuvent parfois être utilisés pour automatiser ou surveiller l'exécution de parties de contrats légaux. Ils peuvent également mettre en œuvre des jeux, des paris ou des loteries (XU et al., 2019).

B.4 Tokenisation

La tokenisation est le processus de conversion d'un actif ou des droits de propriété d'un actif en une unité unique appelée jetons (tokens). Les jetons sont couramment utilisés dans la technologie blockchain pour indiquer la propriété d'un actif de valeur. Les actifs tokénisés peuvent être conçus pour être librement échangeables en ligne et permettre aux investisseurs d'acquérir une propriété fractionnée des actifs. La tokénisation fournit également une représentation numérique de la propriété complète ou partagée des entités,

ils peuvent représenter des actifs tangibles comme l'or, l'art ou l'argent, ou des actifs intangibles comme les droits de vote, les droits de propriété ou les licences de contenu (DEJESUS, 2022 ; STAFF, 2021).

B.5 Actifs numériques

Il existe deux grands types d'actifs numériques : la crypto-monnaie et les jetons. La crypto-monnaie est implémentée directement au noyau de la plateforme de la blockchain, où elle est utilisée comme un mécanisme d'incitation pour les opérations. La blockchain permet la propriété exclusive et des transactions sécurisées de la crypto-monnaie. En revanche, les jetons sont implémentés au-dessus des plateformes blockchain, en utilisant les données de transactions ou les fonctionnalités des contrats intelligents fournies par la blockchain. Par exemple le bitcoin permet aux développeurs d'ajouter 40 octets de données arbitraires à une transaction, qui est ensuite enregistrée de façon permanente sur la blockchain (XU et al., 2019).

B.6 Nonce

En cryptographie, un nonce est un nombre arbitraire qui est généré pour une utilisation spécifique et qui est utilisé pour protéger les communications privées en empêchant les attaques par rejet (replay attacks).