République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Saâd Dahlab – Blida 1
Faculté des sciences

Thèse
Présentée pour l'obtention du **diplôme** de **Master**

En : Informatique

Spécialité : Sécurité des systèmes d'information

Par : KHIALI Mohamed Abdellah et BENNAMANI Amine

Sujet

Détection d'intrusion basé sur l'apprentissage fédéré dans l'internet des objets médicaux

Soutenue publiquement, le 20/09/2022, devant le jury composé de :

Mme.	Narhimene Boustia	Professeur	à Saad Dahlab Blida 1	President
M.	Si-Ahmed Ayoub	Doctorant	à Saad Dahlab Blida 1	Co-directeur
Mme.	Aroussi Sana	Maître de Conférences A	à Saad Dahlab Blida 1	Examinatrice
Mme.	Abed Hafida	Professeur	à Saad Dahlab Blida 1	Examinatrice

Résumé

L'internet des objets médicaux est un ensemble de dispositifs médicaux et d'applications qui se connectent aux système de santé via Internet afin de partager des données sensibles, cela aident l'industrie de santé à avoir de meilleurs contacts et soins envers leurs patients, l'inconvénient de cette technologie réside dans la confidentialité et la sécurité des données, les cyberattaques dans le secteur de la santé ont accéléré au cours des dernières années à cause de l'augmentation du nombre des utilisateurs de l'IoMT dû à l'élévation des patients souffrant de maladies chroniques et des épidémies. De ce fait, dans de nombreux travaux, la sécurisation des applications et des données de santé attire l'attention des chercheurs. L'objectif principal de cette recherche est de développer un système robuste et efficace qui répond aux exigences de sécurité dans l'espace de la santé connectée. Plus précisément, nous développons un mécanisme de détection d'attaque fédéré dans les dispositifs de santé connectés en utilisant un système de détection d'intrusion basé sur l'apprentissage profond (DL) afin de classifier les attaques et améliorer la sécurité de l'IoMT grâce à sa capacité à détecter les attaques zero-day et les nouvelles attaques, et aussi éviter le transfert des données privés entre les périphériques grâce à l'adoption de l'approche de FL . Nos résultats expérimentaux démontrent que notre approche a atteint une précision intéressante, avec un taux d'Accuracy égale a 98% en utilisent UNSW-NB15 comme jeux de données.

mot clées : iomt, Apprentissage Fédéré, Apprentissage Profond, Système de Détection d'Intrusion.

Abstract

The Internet of Medical Things (IoMT) is a set of medical devices and applications that connect to health systems via the internet in order to share sensitive data, this help the health industry to have better contact and care towards their patients, the disadvantage of this technology lies in the privacy and security of data, cyber attacks in the health sector have accelerated in recent years due to the increase in the number of users of the IoMT due to the rise of patients suffering from chronic diseases and epidemics. As a result, in many works, securing health applications and data is attracting the attention of researchers. The main objective of this research is to develop a robust and efficient system that meets the security requirements in the connected health space. Specifically, we develop a federated attack detection mechanism in connected health devices using a deep learning (DL) based intrusion detection system to classify attacks and improve IoMT security through its ability to detect zero-day and novel attacks, and also avoid the transfer of private data between devices through the adoption of the federated learning approach. Our experimental results show that our approach achieved an interesting accuracy, with an Accuracy rate equal to 98% using a UNSW-NB15 Dataset.

Keywords: Internet of Medical Things, Federated Learning, Deep Learning, Intrusion Detection System.

Remerciement

Tout d'abord nous tenons à remercier Dieu, le tout puissant et miséricordieux, qui nous a donné la force, l'intelligence et la patience d'accomplir ce travail. On tient à remercier toutes les personnes qui ont contribué à la réalisation de notre projet. Nos parents, pour leur soutien constant et leurs encouragements. Mme Narhimene BOUSTIA et Mr Ayoub SI-AHMED en tant que directeurs de mémoire, qui nous on guidées dans notre travail et pour leurs remarques, conseils et disponibilité.

Contents

List of Figures

Acronymes

FL Federated learning

ML Machine learning

DL Deep learning

CNN Convolutional Neural Networks

RNN Recurrent Neural Network

DNN Deep Neural Network

HFL Horizontal Federated Learning

VFL Verticak Federated Learning

FTL Federated Transfer Learning

Iot Internet of things

Iomt Internet of medical things

IDS Intrusion detection systeme

TN True negatif

TP True positif

FP False positif

FN False Negatif

Introduction

L'internet des objets (Iot) est un concepte technologique qui fait references a tout les objets qui peuvent etre interconnecté entre eux a l'aide de capteurs et former un reseaux. Ces objets intelligents sont capables de collecter, d'échanger et d'analyser les données, ce qui permet une automatisation de certaines taches comme la surveillance en temps réel des équipements, la gestion des stocks et la traçabilité des produits, et l'amélioration de l'efficacité opérationnelle. L'Iot trouve des applications dans divers domaines, allant des villes intelligentes a la maison intelligente, et avec le développement technologique L'Iot perce aussi dans le domaine médicale communément appelé Internet des Objet médicaux (Iomt).

L'Iomt représente un écosystème interconnecté comprenant des capteurs, des dispositifs portables, des équipements médicaux et des systèmes cliniques. Cet écosystème permet une multitude d'applications dans le domaine de la santé, telles que la surveillance à distance de l'état de santé, les programmes de remise en forme ainsi que la gestion des maladies chroniques [7]. En collectant des données relatives à leur condition physique et en informant les professionnels de la santé en cas de problèmes détectés, ces systèmes contribuent également à garantir le bien-être et à suivre l'évolution de la santé des individus.

L'architecture de l'Iomt se compose de cinq couches principale: la couche perception, la couche réseau, la couche physique, la couche métier et la couche application. La sécurité de la couche physique est récemment apparue comme une alternative à la cryptographie, exploitant les propriétés de la couche physique du système de réseau pour améliorer la

sécurité des systèmes Iomt, sur des réseau public sans fil vulnérable et qui peuvent être exploité par des intrus. De ce fait un système de détection d'intrusion (IDS) est primordial pour la sécurité du réseau car le but principal de ce dernier est d'analyser, détecter et avertir les administrateurs du système [8].

Cependant, l'Iomt présente des challenges. Les préoccupations majeures concernant la sécurité et la confidentialité des données de santé sont que la collecte et le partage de ces données sensibles nécessitent des protocoles de sécurité robustes pour protéger la vie privée et contrée les cyberattaques. Selon une étude faite par le cabinet « Cybersecurity Ventures », le marché mondial de la cybersécurité des soins de santé augmentera de 15% chaque année pour atteindre 125% [9].

Au cours des dernières années, plusieurs propositions de recherche ont été faites pour traiter les vulnérabilités des dispositifs Iomt, parmi celles-ci les solutions d'un IDS basé sur l'apprentissage automatique. Cependant le changement rapide de comportement du réseau et l'évolution de diverses attaque a fait que certaines solutions de l'apprentissage automatique soient l'option la moins préféré en raison de la condition d'avoir un jeux de données chose qui ne peut pas se réaliser dans le domaine de la santé en raison de réglementations prises par certains hôpitaux, aussi le risque de sécurité liés aux transfert de données confidentielles. L'une des approches prometteuses et adaptés à l'iomt est le l'apprentissage fédérer, qui est une technique qui entraine un modèle sur plusieurs appareils décentralisées dotés d'un jeux de données locale, et seuls les poids du modèle appris qui sont transférés à un serveur central ce qui garantit la confidentialité des données des patients.

La contribution dans ce travail peut se résumer comme suit :

— La conception d'un modèle d'apprentissage fédéré qui peut construire un modèle global paramétré et personnalisé dans un serveur central à partir de modèles locaux dans chaque périphérique du réseau connecté.

— Le developement d'un model de ML basé sur le DL.

— Une comparaison des different strategies de l'apprentissage fédéré.

Ce mémoire de fin d'études suit la structure suivante : il commence par les notions de base, puis se penche sur l'apprentissage fédéré et les recherches connexes. Ensuite, il aborde la conception du projet, suivi par les expériences et l'analyse des résultats. Enfin, une conclusion générale récapitulera les points clés et proposera des perspectives d'amélioration futures.

Chapter I

Notions de bases

Dans ce chapitre, nous allons d'abord présenter les notions de base du domaine des objets connectés de manière générale, puis nous nous concentrerons sur son application spécifique dans le domaine médical appelé Iomt. Nous détaillerons ensuite l'architecture utilisée au sein de l'Iomt, les défis auxquels il fait face et les avantages qu'il offre. Enfin, nous aborderons les technologies qui peuvent contribuer à la sécurisation de l'Iomt, notamment ML, qui peut être intégré à un système de détection d'intrusions, ainsi que des concepts plus récents tels que FL.

1.1 Internet des objets

1.1.1 Historique

L'expression "Internet of Things" est attribuée à Kevin Ashton, le directeur exécutif d'Auto-ID Labs. En 1999, il a inventé cette expression en faisant valoir que si nous pouvions ajouter des informations numériques à des objets du monde réel, nous pourrions rapidement et facilement collecter des données sur ces objets et leur environnement [10].

L'iot(Iot) est une technologie émergente qui permettra à des appareils non informatiques de communiquer entre eux via l'infrastructure de réseau existante [11]. L'idée de l'Iot est que tout ce qui possède un interrupteur ou un capteur peut désormais être relié

à l'internet et devenir un nœud de ce réseau mondial. Cela signifie que tous les appareils, peuvent être dotés d'intelligence en étant directement connectés à l'internet.

1.1.2 Architecture de l'Iot

Pour offrir une expérience sans faille de l'iot(Iot), les trois composants nécessaires sont le matériel, les intergiciels (Middleware) et la présentation. Le matériel se compose de capteurs (capteurs de température et de mouvement), d'actionneurs (moteurs) et de matériel de communication embarqué. Les intergiciels pour le stockage à la demande et les outils informatiques pour l'analyse des données comprennent des solutions de stockage en nuage (Cloud) comme Amazon Web Services (AWS) ou Google Cloud Platform (GCP), ainsi que des outils d'analyse de données volumineuses comme Apache Hadoop ou Spark. Les outils de présentation comprennent des outils de visualisation et d'interprétation faciles à comprendre.

L'iot présente trois caractéristiques essentielles. La première est l'instrumentation : l'intégration de capteurs, de puces, d'étiquettes d'identification par radiofréquence (RFID) et de codes à barres dans des objets ordinaires. La deuxième est l'interconnexion : la liaison de ces terminaux entre eux dans un réseau autonome. La troisième est l'intelligence : il s'agit de rendre intelligents les services pervasifs qui utilisent l'ioten leur permettant de réagir aux changements de leur environnement.

Les appareils de l'iotutilisent des microcontrôleurs et des capteurs pour collecter des données à partir de leur environnement, des modules de communication, notamment la RFID utilisée pour l'identification et le suivi des objets, le Bluetooth utilisé pour connecter deux petits appareils entre eux, le ZigBee pour créer des réseaux automatiques entre pairs, les liaisons RF Wi-Fi et les réseaux cellulaires.

L'iot (Iot) est un concept large sans architecture uniforme. Pour que l'Iot soit un succès, il doit être composé d'un assortiment de technologies de capteurs, de réseaux, de communications et de calcul. L'Union internationale des télécommunications (UIT)

propose une architecture de réseau à quatre couches pour l'iot: une couche de détection, une couche d'accès, une couche réseau et une couche intergiciel [12].

1.2 Internet of medical things

Les applications Iot se répartissent en plusieurs catégories, notamment le transport et la logistique, les soins de santé et l'assainissement, les villes intelligentes et l'internet industriel des objets (IIot).

L'iot (IdO) est un concept qui fait référence à des plateformes de dispositifs interconnectés, dotés de capacités web, et à des technologies médicales utilisant des capteurs, des processeurs, des logiciels et du matériel pour capturer, envoyer, analyser et gérer des données. L'iomt (Iomt) est une sous-catégorie de l'iotqui révolutionne l'industrie de la santé.

L'iomt offre de nouvelles opportunités aux fournisseurs de soins de santé pour améliorer les soins aux patients et gérer la complexité inhérente au marché de la santé. Les progrès technologiques permettent de se passer des visites en cabinet. en effet L'Iomt conduit à une révolution dans le secteur de la santé [13].

1.2.1 Architecture de L'Iomt

L'iot medicaux vise à recueillir en temps réel les données de santé des patients et à les stocker dans des infrastructures réseau connectées à Internet. Les médecins utilisent ces données de santé mobile pour surveiller, diagnostiquer et traiter leurs patients. De plus, l'intégration de dispositifs de santé mobiles dans l'environnement des patients permet de prédire en temps réel diverses anomalies lier à la santé [14].

l'architecture de l'Iomt dans la prestation des soins de santé se compose de cinq couches de base : la première couche de perception, la deuxième couche de détection, la troisième couche de réseau, la quatrième couche métier et la cinquième couche d'application.

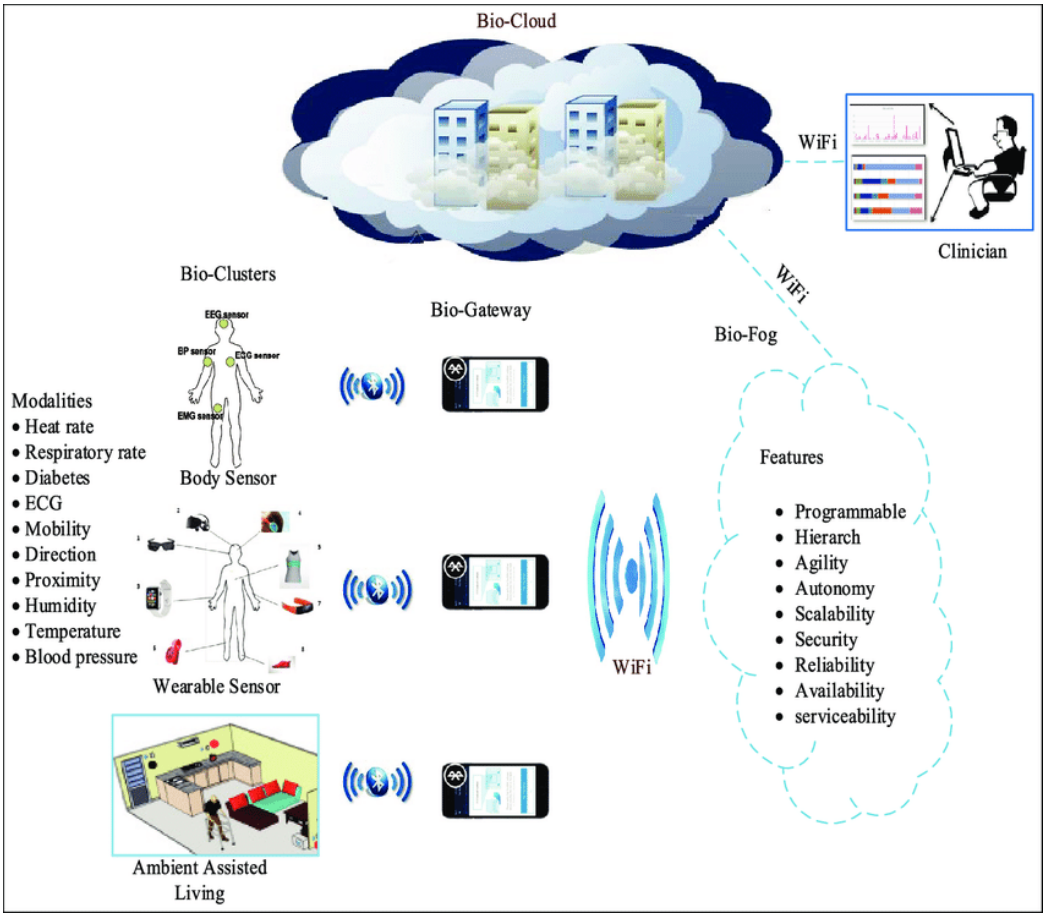


Figure 1.1 – Vue d'ensemble générale de l'architecture Iomt[1]

La couche de perception

Cette couche a pour rôle principal la collecte de données (telles que la fréquence cardiaque, la pression, la température, etc.) en utilisant des capteurs dans la couche physique [15].

La couche de détection

Les applications de santé mobile utilisent une grande quantité de données. Pour stocker, envoyer et recevoir ces données, les dispositifs sont connectés à des serveurs Iot. Les caractéristiques les plus importantes des dispositifs médicaux sont les suivantes : une faible consommation d'énergie, une alimentation en IP et une connectivité sans fil, ainsi qu'une légèreté et une facilité d'utilisation [16].

L'objectif des dispositifs connectés est d'aider à la surveillance, au diagnostic et au traitement des personnes malades, tout en maintenant une faible consommation d'énergie, un coût réduit, une taille physique réduite et une facilité d'utilisation.

La couche de réseau

Cette couche est constituée de systèmes câblés et sans fil, ainsi que de logiciels intermédiaires, qui traitent et transmettent les informations collectées par la couche de perception à l'aide de plates-formes techniques [17].

La couche de metier

Le rôle principal de cette couche est de gérer la logique métier du prestataire de soins médicaux et d'aider au cycle de vie de l'entreprise (par exemple, le contrôle, la surveillance et l'ajustement) des procédures commerciales [18].

La couche d'application

La couche d'application est chargée de convertir ces données dans un format compréhensible par les dispositifs finaux et les serveurs médicaux [19].

1.2.2 Challenges de l'Iomt

Iomt est un domaine qui a vu le jour récemment, donc comme chaque nouveau domaine, il doit forcément faire face à certains challenges, parmi ces challenges :

Securite

En générale, l'iotimplique la gestion de grandes quantités de données provenant de capteurs. Lorsqu'il s'agit de l'Iot dans les dispositifs médicaux, ces données sont souvent hautement sensibles, comprenant des informations sur l'état et la localisation d'une personne malade, des détails sur le traitement, et bien plus encore [20].

Connectivite

N'importe quel system Iomt qui est supposé gérer des données en temps réel doit contenir une connectivite stable et non interrompu en tout temps.

Interoperabilite des données

I'Iomt ne marche pas tres bien sans des données correcte, et ce n'est pas choquant vu que une des clées primaires de cettes technologie et que ca aide dans la prise de decision.

Standardisation

Le manque de standardisation a un impact sur l'interopérabilité des dispositifs médicaux, ce qui réduit l'efficacité globale de l'Iomt [21].

1.3 Système de détection d'intrusion

Les chercheurs ont mis en évidence diverses attaques contre les dispositifs médicaux, notamment l'écoute clandestine, l'altération des messages, l'injection de fausses données et les attaques par déni de service, qui peuvent compromettre la sécurité des patients, la sûreté et la disponibilité des systèmes critiques [22].

Un système de détection d'intrusion (IDS), est un système conçu pour détecter ou repérer des comportements suspects ou anormales sur un réseau ou une machine. Son rôle principal est de donner un rapport complet sur toutes tentatives d'intrusion, un signal d'alerte sera déclenchée lors d'une tentative réussie ou échouée détecté sur la cible surveillée [23].

Les IDS se divisent généralement en deux grandes catégories, les plus populaires étant la détection basée sur les signatures et la détection des anomalies qui est souvent associée à ML. Certains IDS sont capables de réagir aux menaces détectées, ce qui en fait des systèmes de prévention d'intrusion réactifs[24] [25].

1.3.1 Systèmes de détection d'intrusion par signatures

Les systèmes de détection d'intrusion basés sur les signatures (SIDS) fonctionnent en utilisant des biblIothèques de signatures. Lorsqu'ils analysent les flux réseau, ces systèmes examinent chaque transmission et envoient un message en cas de comportement anormal. Cependant, cette approche a ses limites car elle dépend de l'identification préalable des différentes attaques et nécessite des mises à jour régulières de la base de signatures. Par conséquent, ce modèle de détection se révèle inefficace lorsqu'il s'agit d'attaques non répertoriées dans la base de signatures [25].

1.3.2 Systèmes de détection d'intrusion par anomalies

Les systèmes basés sur les anomalies, à la différence des SIDS, ne dépendent pas de biblIothèques, mais sont conçus pour détecter automatiquement les comportements anormaux grâce à l'apprentissage. Ces systèmes passent par deux phases [25] [24] :

— La première phase est l'apprentissage, où le système étudie le comportement normal des flux réseau.

— La seconde phase est la détection, où le système analyse le trafic et essaie d'identifier les événements anormaux en se basant sur les connaissances acquises. Cette méthode de détection utilise différentes techniques d'apprentissage automatique.

D'aussi les que les SIDS ne sont pas capable de detecter les attaques zero-day, tandis que les AIDS peuvent le faire, par contre ca peut generer un taux augmenter de faux-positive car les animalies peuvent etre just de nouvelles activités et non des intrusion [25] [24]

1.4 Apprentissage automatique

ML est une technologie de l'intelligence artificiel, appeler aussi ML, son but principale est de donner la capacité aux machines d'apprendre sans avoir été au préalablement programmées spécialement à cet effet. En se basant sur un forage de données elle doit en tirer des schéma ou des modèles pour donner des prédictions en se basant sur des statistiques. Donc un programme informatique traditionnel suit des instructions précises pour faire une tâche tandis qu'un système ML ne suit pas d'instructions mais apprend et améliore ses performances avec le temps [26].

1.4.1 Types d'apprentissage automatique

ML se compose de trois types qui sont [27] :

Apprentissage non-supervisé

Dans ce type en doit entrainer le modèle sur des données non-étiqueter, la machine parcourt les données et essaye de trouver des points en commun [26].

Apprentissage supervisé

Est une tâche consistant à apprendre une fonction de prédiction à partir d'exemples annotés [26].

Apprentissage semi-supervisé

C'est la combinaison d'une petite quantité de données étiquetées avec une grande quantité de données non étiquetées durant l'entrainement. Il est souvent utilisé pour modifier ou réorganiser les hypothèse effectuées sur le modèle [26].

1.4.2 Fonctionnement du ML

Le fonctionnement de l'apprentissage automatique repose sur un modèle spécifique développé en quatre étapes qui sont généralement gérées et suivis par un data scientist.

En premier, on doit sélectionner un ensemble de données pour l'entraînement, son but principal est de nourrir le modèle de ML afin d'apprendre à résoudre le problème visé, ces données peuvent être étiquetées ou non, l'étiquète est faite pour faciliter au modèle les caractéristiques qu'il devra identifier. Dans les deux cas, les données doivent être bien préparées et nettoyées sinon l'entrainement du modèle risque d'être non fiable et par la suite ses prédictions seront faussées. Ensuite, on doit sélectionner un algorithme à exécuter sur les données d'entrainement, dans cette étape plusieurs facteurs sont prises en considérations comme le type et le volume de données d'entrainements ainsi que le type de problème à résoudre. La troisième étapes est répétitif, on parle de l'entrainement de l'algorithme sélectionner, les variables de ce dernier sont exécutées à chaque itérations puis le résultat est comparé avec ceux qu'il aurait dû produire. Ce processus est réalisé

dans le but d'augmenter la précision du résultat de l'algorithme. Enfin la dernière étape est consacré à l'entrainement et l'amélioration du modèle de ML en utilisent de nouvelles données qui ont un rapport avec le problème à résoudre [26].

1.5 Apprentissage profond

C'est un sous domaine de ML et une combinaison de la puissance informatique et les réseaux de neurones artificiels afin d'apprendre des patterns complexes au seins de large quantités de données, par exemple identifier un concert dans un Imagerie par résonance magnétique (IRM) ou un mot dans les sons. Ce type est inspiré par la façon dont le cerveau (plus exactement les neurones naturels) traite l'information en réagissant aux stimuli, il est caractérisé par l'effort de créer un modèle a plusieurs niveaux, les plus profonds prennent en considération les résultats des prédécesseurs, les réseaux de neurones artificiels font partie des moyens qui rendent les machines plus humains en les aidant à raisonner comme eux [26].

1.5.1 Reseaux de neurones

Ces systèmes sont constitués d'un ensemble de neurones artificiels ou formels qui créent un graphe orienté pondéré [28]. Chaque neurone possède un état interne et une fonction d'activation qui affectent les autres neurones. L'activité se propage à travers le graphe via des liens synaptiques pondérés [29].

Le neurone artificiel

Chaque neurone artificiel est un opérateur mathématique, on peut dire processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w (weight : poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque

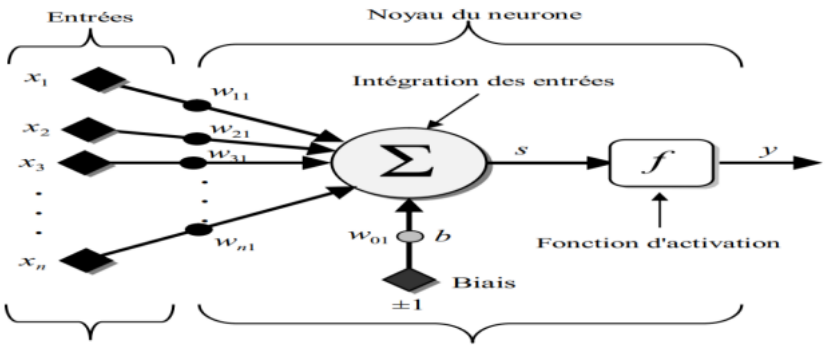


Figure 1.2 – Modèle d'un neurone artificiel [2]

connexion est associé un poids [30]. On a pris l'habitude de représenter graphiquement un neurone artificiel comme indiqué sur la figure 1.2.

— **xi** représentent les vecteurs d'entrées.

— Les **wij** sont les poids synaptiques du neurone j.

— **Biais** : entrée prend souvent les valeurs -1 ou +1 qui permet d'ajouter de la flexibilité auréseau en permettant de varier le seuil de déclenchement du neurone par l'ajustement des poids et du biais lors de l'apprentissage.

— **Noyau** : intègre toutes les entrées et le biais et calcul la sortie du neurone selon unefonction d'activation qui est souvent non linéaire pour donner une plus grande flexibilité d'apprentissage.

Les couches

Un réseau de neurones est un ensemble de neurones formels interconnectés, organisés en trois couches : la couche d'entrée (input layer), la couche cachée (hidden layer) et la couche de sortie (output layer). Chaque couche est composée de un ou plusieurs nœuds, représentés par des petits cercles dans le diagramme. Les connexions entre les nœuds indiquent la propagation de l'information d'un nœud à un autre. Dans ce type particulier de réseau de neurones, les informations ne se déplacent que de la couche d'entrée vers la

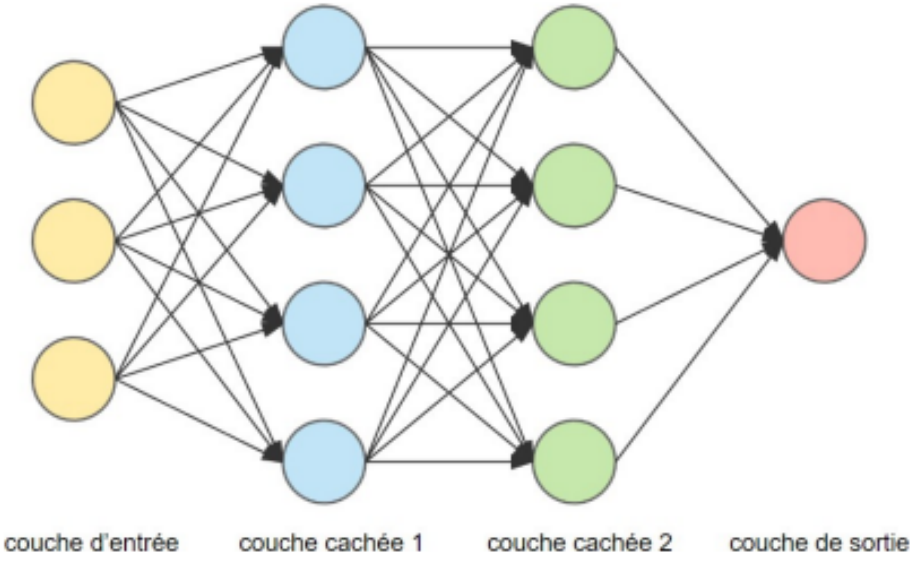


Figure 1.3 – Modèle d'un reseaux de neurones [3]

couche de sortie, voir la figure 1.3.

1.6 Apprentissage Fédéré

Le ML et le DL sont des techniques d'apprentissage automatique qui permettent aux ordinateurs d'apprendre à partir de données. Cependant, ces techniques ont échoué à faire face à des problèmes de confidentialité et de sécurité liés à la centralisation des données. Le FL a fait un pas en avant en permettant de former un algorithme via plusieurs sessions indépendantes, chacune utilisant son propre ensemble de données [31] [32].

FL est une technique d'apprentissage automatique qui entraîne un algorithme sur plusieurs appareils sans partager les données entre les appareils. Cela est réalisé en permettant à chaque appareil de former un modèle local sur ses propres données, puis en échangeant les mises à jour des paramètres du modèle avec un serveur central. Le serveur central agrège ensuite les mises à jour de tous les appareils et met à jour le modèle global. Ce processus est répété jusqu'à ce que le modèle global converge.

FL présente plusieurs avantages par rapport aux techniques traditionnelles d'apprentissage

automatique qui nécessitent une centralisation des données. Premièrement, il permet de protéger la confidentialité des données, car les données ne quittent jamais les appareils. Deuxièmement, il peut être utilisé pour former des modèles sur des données trop volumineuses ou trop sensibles pour être centralisées. Troisièmement, il peut être utilisé pour former des modèles sur des données réparties sur un grand nombre d'appareils [33] [34].

1.6.1 Applications de FL

FL peut être adopté dans de nombreux domaines, notamment les suivants :

la santé

FL peut être utilisé pour former des modèles destinés à des applications de santé telles que le diagnostic des maladies, la découverte de médicaments et la médecine personnalisée. Par exemple, l'étude Google Health a utilisé FL pour former un modèle permettant de prédire le risque d'insuffisance cardiaque chez les patients [35].

Les services financier

FL peut être utilisé pour former des modèles destinés aux applications des services financiers, telles que la détection de fraudes, l'évaluation des risques et la segmentation des clients. Par exemple, le projet de classement personnalisé préservant la vie privée de Google a utilisé FL pour former un modèle permettant de recommander des produits aux utilisateurs sans partager leurs données personnelles [36].

Les equipement de L'internet des objets

FL peut être utilisé pour former des modèles à partir des données provenant des dispositifs Iot tels que les capteurs et les actionneurs. Par exemple, le projet de santé sans fil fédéré a utilisé FL pour former un modèle permettant de prédire le risque de chute chez les personnes âgées en utilisant des données provenant de capteurs portables [37].

1.7 Conclusion

En conclusion, ce chapitre a abordé les concepts fondamentaux du domaine des objets connectés, en mettant l'accent sur leur application spécifique dans le domaine médical, connu sous le nom d'Iomt. Ce chapitre nous a permis d'acquérir une compréhension approfondie de l'état actuel, l'architecture utilisée au sein de l'Iomt, les défis auxquels il est confronté, et les avantages qu'il présente. De plus, nous avons exploré les technologies qui peuvent contribuer à la sécurisation de l'Iomt, telles que ML et FL qui serviront de base solide pour la suite de notre étude.

Chapter II

Apprentissage Fédéré

2.1 Introduction

Dans le domaine de l'intelligence artificielle, les données constituent la base, par conséquent, l'entraînement du modèle ML ne peut être effectué sans les données. Cependant, les données existent souvent sous la forme d'îlots de données (data islands) et ces données ne peuvent etre traiter que de maniere centralisé. Mais dans ce cas la, il peut y avoir des fuite de données ou subir des attaques lors de l'envoie des données au serveur afin d'entrainer un model de ML, alors pour eviter la perte de données et preserver la confidentialité de l'utilisateur, il existe une technique qui consiste a inverser le processus, c'est a dire au lieu d'envoyer les données vers le model, on envoie un model au dispositifs afin de l'entrainer localement puis le renvoyer au serveur central. cette technique s'apelle l'aprentissage fédéré.

2.2 Principe de l'apprentissage fédéré

Comme le montre la figure 2.1, le processus de l'apprentissage fédéré comprend les étapes clés suivantes [4]:

— Initialisation du system : le serveur sélectionne un sous-ensemble de clients qui

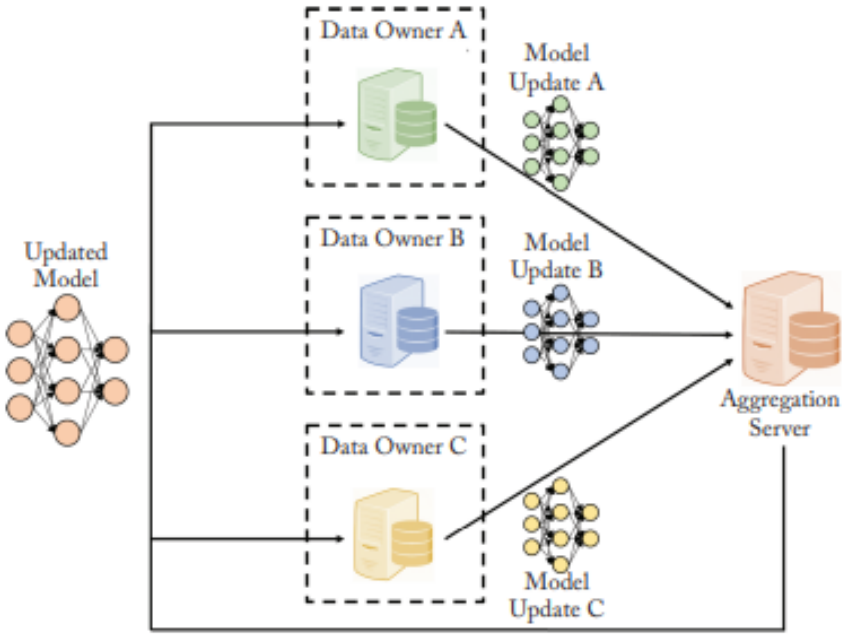


Figure 2.1 – Vue d'ensemble générale de l'architecture FL

devraient être impliqués dans le processus FL, et collection les mise a jours des poids faites par les dispositif localement pour faire une misa a jour du models global et le renvoyer au participants une autre fois

— Distribution local de l'entrainement et mise a jour : une fois l'ensembles de participant est selectionné, le serveur envoie un model global initial a chaque client, puis chaque dispositifs participants effectues plusieurs iterations avec son propre jeu de données local, Ensuite, chaque client envoie la mise à jour de son modèle vers le serveur pour l'agrégation.

— Aggregation du modele : apres avoir recu tous les mise a jours des clients selectionnées, le serveur fait l'aggregation de ces derniers a en utilisant quelque methodes d'aggregations afin de produire une nouvelle version du model global et la partager avec tous les clients, ce processus se repete jusqu'a avoir une version optimiser du model.

2.3 Categories de l'apprentissage fédéré

L'apprentissage fédéré se décline en trois approches, voir la figure 2.2 :

Federated Learning Horizontal (HFL)

Dans le HFL, les clients du Iomt peuvent participer à l'entraînement d'un modèle global partagé en utilisant leurs ensembles de données qui possèdent le même espace de caractéristiques mais des espaces d'échantillonnage différents. [38]

Federated Learning Vertical (VFL)

VFL intervient lorsque les clients sont exposés à des espaces de caractéristiques différentes mais à un espace d'échantillonnage similaire ou identique. les échantillons qui se chevauchent parmi les données du client sont trouvés en utilisant la technique d'alignement des entités, ces données superposées sont utilisées pour la formation[39].

Federated Transfer Learning (FTL)

Contrairement aux systèmes VFL, le FTL est utilisé pour gérer des ensembles de données avec des espaces d'échantillonnage et des espaces de caractéristiques différents. En utilisant une méthode de transfert d'apprentissage, les valeurs des caractéristiques sont calculées à partir de différents espaces de caractéristiques pour obtenir une représentation commune qui est exploitée pour entraîner les ensembles de données locaux [40].

2.4 Les defis de l'apprentissage fédéré

Bien que l'apprentissage fédéré est tres prometteur et qui a un grand potentiel, il est également confronté à plusieurs défis.

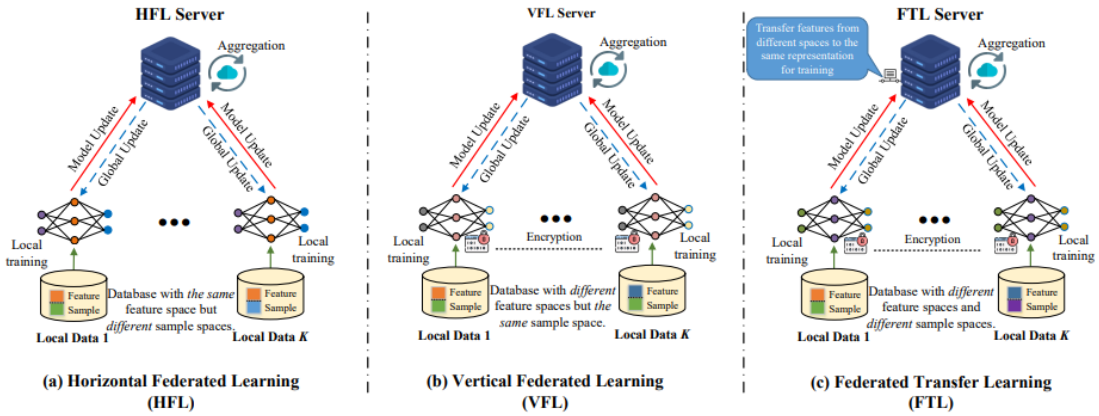


Figure 2.2 – Les categories de l'apprentissage fédéré [4]

2.4.1 Communication coûteuse

Dans les réseaux fédérés, la communication représente un obstacle majeur qui limite considérablement la fluidité des échanges.[41]. Les préoccupations relatives à la protection de la vie privée liées à l'envoi de données brutes nécessitent que données générées sur chaque dispositif restent locales. et vu que les reseaux fédéré comprend potentiellement un nombre massive de dispositifs, cela peut rendre le travaille du reseau un peu plus lent a cause de different facteurs comme le debit, l'energie et la puissance [42], alors pour adapter un modèle avec les données des dispositifs, il est donc important de développer des méthodes efficaces en termes de communication qui envoient itérativement de petits messages ou des mises à jour du modèle, plutôt que d'envoyer l'ensemble des données sur le réseau, et pour reduire d'avantage la communication il faut : (i) réduire le nombre total de communications (ii) réduire la taille des messages transmis à chaque tour.

2.4.2 Hétérogénéité des systèmes

En raison des variations matérielles (CPU et mémoire), les dispositifs au sein des réseaux fédérés peuvent présenter des différences dans leurs capacités de stockage, de

calcul et de communication et de connectivité aux réseau (3G, 4G, 5G et Wi-Fi) aussi a la puissance (niveau de batterie) [42]. En outre, la taille du réseau et les contraintes liées aux systèmes sur chaque dispositif entraîne généralement seulement une petite fraction des dispositifs étant actifs à la fois, par exemple, des centaines de dispositifs actifs dans un réseau de millions d'appareils [41]. Il n'est également pas rare pour qu'un dispositif actif soit abandonné à une itération donnée en raison de contraintes de connectivité ou d'énergie [41]. Ces caractéristiques au niveau du système exacerbent considérablement les défis tels que les retardataires l'atténuation et la tolérance aux fautes. L'apprentissage fédéré les méthodes doivent donc (i) prévoir une faible participation, (ii) tolérer un matériel hétérogène (iii) être robustes assez pour faire tomber des appareils dans le réseau de communication.

2.4.3 Hétérogénéité statistique

Les appareils génèrent et recueillent souvent des données d'une manière très peu identique à l'échelle du réseau, par exemple, les utilisateurs de téléphones mobiles utilisent des termes variés dans le contexte. Ce paradigme de génération de données viole fréquemment des hypothèses indépendantes et identiquement distribuées (i.i.d.) dans l'optimisation distribuée et peut ajouter de la complexité en termes de la modélisation des modeles, l'analyse théorique et l'évaluation empirique des résultats. En effet, bien que le problème canonique d'apprentissage fédéré vise à apprendre un seul modèle mondial, il existe d'autres alternatives telles que l'apprentissage simultané de modèles locaux distincts via configurations d'apprentissage multitâches [43]. À cet égard, il existe également un lien étroit entre les approches apprentissage fédéré et méta- apprentissage [44]. Les perspectives de multitâches et de méta- apprentissage permettent une modélisation personnalisée ou spécifique à des dispositifs, ce qui est souvent plus approche naturelle pour gérer l'hétérogénéité statistique de les données pour une meilleure personnalisation.

2.4.4 Problèmes de confidentialité

Enfin, la protection de la vie privée est souvent une préoccupation majeure dans l'apprentissage fédéré. L'apprentissage fédéré fait un pas en avant dans la protection des données générées sur chaque appareil en partageant les mises à jour des models. Par exemple, l'information étagée plutôt que les données brutes.Toutefois, la mise à jour des modèles tout au long du processus de formation peut révéler des renseignements de nature délicate, soit pour Serveur tiers ou central. Malgré les méthodes modernes améliorer de l'apprentissage fédéré pour la protection de la vie privée, ces méthodes penchent souvent vers la protection de la vie privée aux dépit du faible rendement du modèle ou efficacité du système. Comprendre et équilibrer ces compromis est un challenge considerable pour l'apprentissage fédéré [42].

2.5 Stratégies l'apprentissage fedéré

l'apprentissage fédéré est fait pour optimiser l'entrainement des models a travers multiple dispositifs avec leurs données tout en gardant les données privé et preserver la confidentialite, ceci est utiliser a l'aide de certaines strategies qui sont :

2.5.1 FedAvg

c'est la strategie la plus commune parmis les autres strategies, dans le FedAVg, chaque clients entraine son modal localement en utilisant son propre data, apres il aggrege les mise a jours des models des clients, par prendre la moyenne des poid des models, cela aggrege le model pour etre utiliser par le model global pour le prochain tours d'entrainement.

Listing II.1 – FedAvg Algorithm

```
# Initialize global model parameters W
W = InitializeGlobalModelParameters()

# for each communication round t do:
for t in range(num_communication_rounds):
    # Sample a fraction of clients C_t
    C_t = SampleFractionOfClients()

    # for each client c in C_t do:
    for c in C_t:
        # Send current global model W to client c
        SendGlobalModelToClient(c)

        # c performs local model update W_c using its local data
        W_c = LocalModelUpdate(c)

    # Aggregate the local model updates from all clients
    W_new = AggregateLocalModelUpdates(C_t)

    # Update the global model
    W = W_new
```

2.5.2 FedAvgM

cette strategie est similaire au fedAVg mais fedAvgM utilise un momentum sur le serveur pour faire la mise a jour du model global, pour le but de stabiliser la convergence est la rendre plus stable.

Listing II.2 – FedAvgM Algorithm

```
# Initialize global model parameters W and momentum buffer V
    W, V = InitializeGlobalModelParameters()

# for each communication round t do:
for t in range(num_communication_rounds):
    # Sample a fraction of clients C_t
    C_t = SampleFractionOfClients()

# for each client c in C_t do:
    for c in C_t:
    # Send current global model W and momentum buffer V to client c
        SendGlobalModelAndMomentumToClient(c, W, V)

    # c performs local model update W_c and updates its local momentum buffer
        V_c
        W_c, V_c = LocalModelUpdateWithMomentum(c, W, V)

# Aggregate the local model updates and momentum buffers from all clients
    W_new, V_new = AggregateLocalModelUpdatesWithMomentum(C_t, W_c, V_c)

# Update the global model and momentum buffer
    W, V = W_new, V_new
```

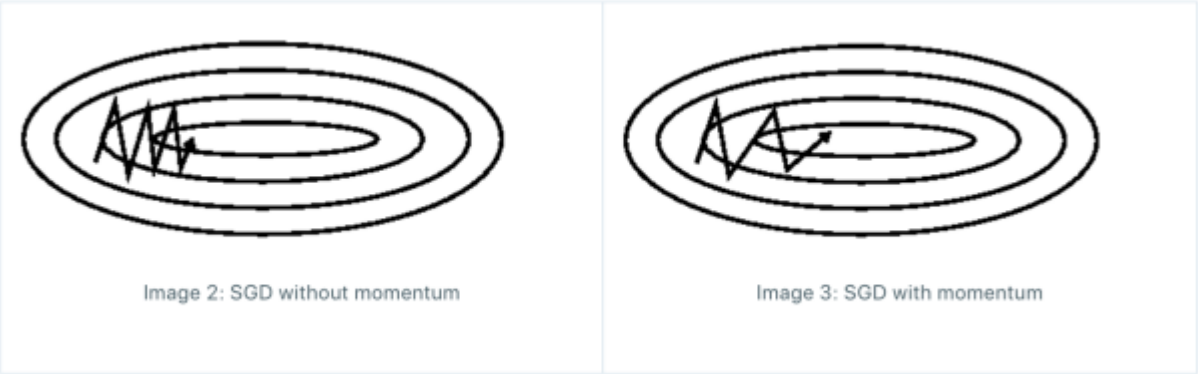


Figure 2.3 – – Représente comment convergence un SGD avec momentum et un SGD sans momentum [5]

2.5.3 FedYogi

FedYogi est une strategie FL basé sur l'optimiseur Yogi, qui est un algorithme de la gradiant stochastic, il combine le benefice de l'apprentissage adaptive avec la convergence de la methode traditionel SGD.

Listing II.3 – FedYogi Algorithm

```
# Initialize global model parameters W
W = InitializeGlobalModelParameters()

# for each communication round t do:
for t in range(num_communication_rounds):
    # Sample a fraction of clients C_t
    C_t = SampleFractionOfClients()

    # for each client c in C_t do:
    for c in C_t:
        # Send current global model W to client c
        SendGlobalModelToClient(c)

        # c performs local model update W_c using its local data
        W_c = LocalModelUpdate(c)

    # Aggregate the local model updates from all clients
    W_new = AggregateLocalModelUpdates(C_t)

    # Update the global model using Yogi optimizer
    W = YogiOptimizer(W, W_new, beta1, beta2, epsilon)
```

2.5.4 FedAdagrad

dans cette strategies chaque client adapte son taux d'aprentissage en se basant sur la distribution de ses données local, pour le but d'avoir une meilleur convergence, speciale-ment quand les client ont des données non IID.

Listing II.4 – FedAdagrad Algorithm

```
# Initialize global model parameters W
W = InitializeGlobalModelParameters()

# for each communication round t do:
for t in range(num_communication_rounds):
    # Sample a fraction of clients C_t
    C_t = SampleFractionOfClients()

    # for each client c in C_t do:
    for c in C_t:
        # Send current global model W to client c
        SendGlobalModelToClient(c)

        # c performs local model update W_c using its local data
        W_c = LocalModelUpdate(c)

    # Aggregate the local model updates from all clients
    W_new = AggregateLocalModelUpdates(C_t)

    # Update the global model using Adagrad optimizer
    W = AdagradOptimizer(W, W_new, learning_rate)
```

2.5.5 FedAdam

FedAdam combine le taux d'apprentissage adaptive avec son apprentissage fédéré, c'est a dire qu'il adapte le taux d'apprentissage en se basant sur le premier et second moment du gradients.

Listing II.5 – FedAdam Algorithm

```
# Initialize global model parameters W and exponential moving average buffers M
    and V
W, M, V = InitializeGlobalModelParametersAndBuffers()

# for each communication round t do:
for t in range(num_communication_rounds):
    # Sample a fraction of clients C_t
    C_t = SampleFractionOfClients()

    # for each client c in C_t do:
    for c in C_t:
        # Send current global model W and exponential moving average buffers M
            and V to client c
        SendGlobalModelAndBuffersToClient(c, W, M, V)

        # c performs local model update W_c and updates its local buffers M_c and
             V_c
        W_c, M_c, V_c = LocalModelUpdateWithAdam(c, W, M, V)

    # Aggregate the local model updates and buffers from all clients
    W_new, M_new, V_new = AggregateLocalModelUpdatesWithAdam(C_t, W_c, M_c, V_c)

    # Update the global model and exponential moving average buffers
    W, M, V = AdamOptimizer(W, W_new, M, M_new, V, V_new, learning_rate, beta1,
        beta2, epsilon)
```

2.6 Travaux connexes

Plusieurs articles ont abordé la sécurité dans l'Iomt, la plupart d'entre eux ont données des solutions de l'approche machine learning traditionnelles, ou des solutions sur l'approche de FL mais on se basant seulement sur la théorie sans aucune expérience pratique comme par exemple les recherches menés par **V.Patel et al.** Dans [45] ou les auteurs ont proposé le FL comme solution pour surmonter le problème de confidentialité des données dans les systèmes de santé Iomt. Cet article traite une enquête complète sur l'émergence de FL comme solution pour soutenir le domaine de la santé, il parle aussi des modèles FL émergents et propose une architecture informatique de santé en couches basées sur FL. Les auteurs ont présenté une analyse détaillée de la catégorisation FL et des algorithmes d'agrégation. Cependant, ils n'ont discuté d'aucun scénario de cas d'utilisation ni d'applications.

Alors que dans les travaux de **M.Ferrag et al.** Dans [46], ils ont présentés un examen des système de sécurité et de confidentialité basée sur l'apprentissage fédéré pour plusieurs type d'applications Iot, notamment le domaine de la santé, ensuite ils ont parlé des vulnérabilité des systèmes de sécurité et de confidentialité sur l'apprentissage fédéré, enfin ils ont réalisé une analyse expérimentale en comparant les performances de trois approches d'apprentissage centralisé, à savoir les réseaux de neurones récurrents (RNN), les réseaux de neurones convolutifs (CNN) et les réseaux neuronaux profonds (DNN), avec celles de l'apprentissage fédéré. L'expérience a été faite avec trois jeux de données, le premier est l'ensemble de données Bot-Iot [47], le second MQTTset [48] et le dernier TON-Iot [49]. Le but de cette expérience est de démontrer que les approches d'apprentissage en profondeur fédéré garantissent une plus grande précision dans la détection des attaques et la confidentialité des données que les approches d'apprentissage automatique centralisé. Les résultats obtenus démontrent la précision des techniques d'apprentissage en profondeur (DNN, CNN et RNN), dans les ensembles de données Bot-Iot [47] et TON-Iot [49], la

précision la plus élevée est obtenu avec le RNN qui atteint 96,76% et 99,98% successivement, et pour l'ensemble de données MQTTset la précision la plus élevé était pour le classificateur DNN avec 90,06%. Tandis que dans l'apprentissage en profondeur fédéré et après 50 rounds d'apprentissage les performances du modèle globale ont réussi à se rapprocher des performance du modèle centralisé. En conclusion les modèles centralisé ont une grande précision de détection d'attaques, cependant il y a deux problèmes à savoir un problème de confidentialité et de sécurité. Le premier vient du fait que il existe une seule entité de collecte de données ce qui permet à un attaquant de la cibler pour faire tomber tout le systéme, et deuxièmement étant donné l'énorme flux de données transférés par les appareils vers l'unité centrale unique, on aura des problèmes de traitement, et ces problèmes sont résolu avec la détection d'intrusion basé sur l'apprentissage fédéré.

Dans les travaux menés par J.B.Awotunde et al. Dans [50], ils ont proposé un modèle basé sur un réseau de neurones en essaim pour un IDS avec une architecture de données centralisé afin de détecter les intrusions pendant le transfert de données dans un l'iomtde santé, le modèle a été testées à l'aide d'un jeux de données NF-ToN-Iot [51] puis ils ont comparés ses résultats aux modèles de classifications de détection standard avec la même base de données, la simulation du modèle proposé a donnée de bon résultats avec une précision de 89%, Dans cette solution les données sont analyser seulement au niveau des objets connectés, de ce fait les données privés des patients sont transférées d'un périphérique a un autre sans aucunes préventions, Par conséquent, la confidentialité des données est exposée aux acteurs malveillant, ensuite ce penché vers une architecture centralisé de données peut causer des problèmes par exemple dans le cas où l'unité d'analyse centrale tombe en panne tout le IDS est inutilisable.

Le travail de **Y.Otoum et al.** dans [52] a apporter une nouvelle solution qui consiste a utilisé un IDS basé sur l'apprentissage de transfert fédérer dans le but de sécurisé les données privées des patients dans le domaine de santé, et assurer une transmission de données sécuriser entre les appareils connectés, la solution apporter est basé sur l'algorithme de réseau neuronal profond (DNN) afin d'entrainer les modèles sur chaque objets con-

nectés et les transférer après au serveur central pour créer un modèle global agrégé sans exposé les données privés des patients, ils ont utilisé CICIDS2017[53] comme jeux de données, cette dernière comporte des paquets réels ainsi qu'un trafic réseau qui contient des attaques courantes distribuées avec le trafic pour différents jours. L'évaluation de leur modéle est basé sur la précision, le taux de détection, le temps moyen de formation et de prédiction, après simulation de quatre scénarios les résultats montrent que la combinaison des ensemble données CICIDS2017-vendredi et CICIDS2017-lundi pour entrainer le modèle global et l'utiliser comme modèle pré-entrainer pour les autres jeux de données donne les meilleurs résultats en terme de performances, une comparaison aussi a été faite avec les méthodes d'apprentissage centralisées traditionnelles avec les même jeux de données et le model proposé a donnée de meilleur résultat, cepandant les techniques de compression peuvent causer des erreurs dans les données, ce qui peut avoir un impact négatif sur les performances du modèle. De plus, appliquer le transfert d'apprentissage à certains problèmes peut s'avérer difficile.

Tandis que **M. Aledhari, et all (2020)** Dans [54] propose un mecanisme de detection d'intrusion pour les dispositif Iot appéllé FedACNN, c'est un mecanisme qui utilise le FL pour un entrainer des model de CNN sur un jeu de données local, afin d'assurer la precision du model et reduire les risques de perte de données. car ils trouvent que l'apprentissage fédéré peut etre prometteur mais aussi peut être coûteux en termes de calcul, car chaque appareil doit former son propre modèle. Les auteurs considerent que les CNN sont un outil puissant pour la détection des intrusions dans les systèmes Iot. donc ils ont évalué FedACNN sur un ensemble de données réelles de trafic Iot NSL-KDD [55]. les résultats ont montré que FedACNN peut atteindre une précision de classification élevée de 99,76%, tout en réduisant la surcharge de communication entre les périphériques et le serveur central, par contre la solution proposée n'est pas adaptée à toutes les applications IoT et nécessite un serveur cloud centralisé.

Nom de l'article	**Méthode Utilisée**	**Avantages**	**Inconvénients**	**Taux de Réussite**
V.Patel et al. [45]	FL	La méthode proposée améliore les performances de machine learning	Pas de scénarios d'utilisation ni d'applications discutés	97,8%
M.Ferrag et al. [46]	DNN CNN RNN	— Amélioration de la confidentialité des données — Amélioration de la sécurité des appareils — Amélioration des performances	— Complexité et probleme d'heterogenité — Peut nécessiter des ressources importantes	96.76% (RNN)
J.B.Awotunde et al. [50]	Modèle basé sur réseau de neurones en essaim pour la détection d'intrusion	Bonne précision	Confidentialité des données exposée, problème en cas de panne du serveur central	89%
Y.Otoum et al. [52]	IDS avec FTL	Sécurité des données privées, transmission sécurisée	Appliquer le transfert d'apprentissage à certains problèmes peut s'avérer difficile. Les techniques de compression peuvent causer des erreurs dans les donnée	Variable
M. Aledhari et al. [54]	Mécanisme FedACNN pour la détection d'intrusion Iot	Haute précision, réduction de la surcharge de communication	La solution proposée n'est pas adaptée à toutes les applications IoT et nécessite un serveur cloud centralisé	99.76%

Table 2.1 – Resume des articles sur la sécurité du FL dans l'IoMT

2.7 Conclusion

Au cours de ce chapitre, nous avons sondé en profondeur les opportunités mises en lumière par l'apprentissage fédéré dans le contexte de l'Iot. Initialement, nous avons éclairci ses principes fondamentaux, puis exploré les différentes catégories qui le composent ainsi que les défis inhérents. Enfin, nous avons étudié les travaux connexes relatifs à la détection d'intrusions au sein de l'Iot en utilisant l'apprentissage fédéré. L'objectif de cette exploration est de façonner un modèle d'IDS basé sur l'apprentissage fédéré, en proposant une approche novatrice et économique pour la détection d'intrusions au sein de l'Iomt. Cette méthode non seulement garantit l'efficacité de la détection, mais préserve également la confidentialité des utilisateurs de manière proactive.

Chapter III

Conception et Implementation

3.1 Introduction

Dans ce chapitre, nous allons développer un système de détection d'intrusion basé sur l'apprentissage fédéré pour l'Iomt systéme en utlisant UNSW-NB15 comme jeu de données, et nous allons aussi detaillé les point important que comporte notre programmes [6].

3.2 Architecture du réseau

Le réseau est composé de dispositifs médicaux sans fil placés sur le corps du patient (qui contient des capteure corporels) et de dispositifs mobiles agissant en tant que passerelle locale pour les dispositifs médicaux, et d'un serveur en arrière-plan à l'hôpital. ces derniers se connectent entre eux en utilisant une communication sans fil a courte porté.

Le système suit une topologie client-serveur entre les dispositifs mobiles des patients et le serveur de l'hôpital de facon que (voir Figure 3.1) :

— Les capteurs sont utilisés pour collecter les paramètres vitaux du patient et injecter des médicaments tels que l'insuline ou les anesthésiques.

— Le dispositif mobile collecte et conserve l'historique des mesures des capteurs, telles

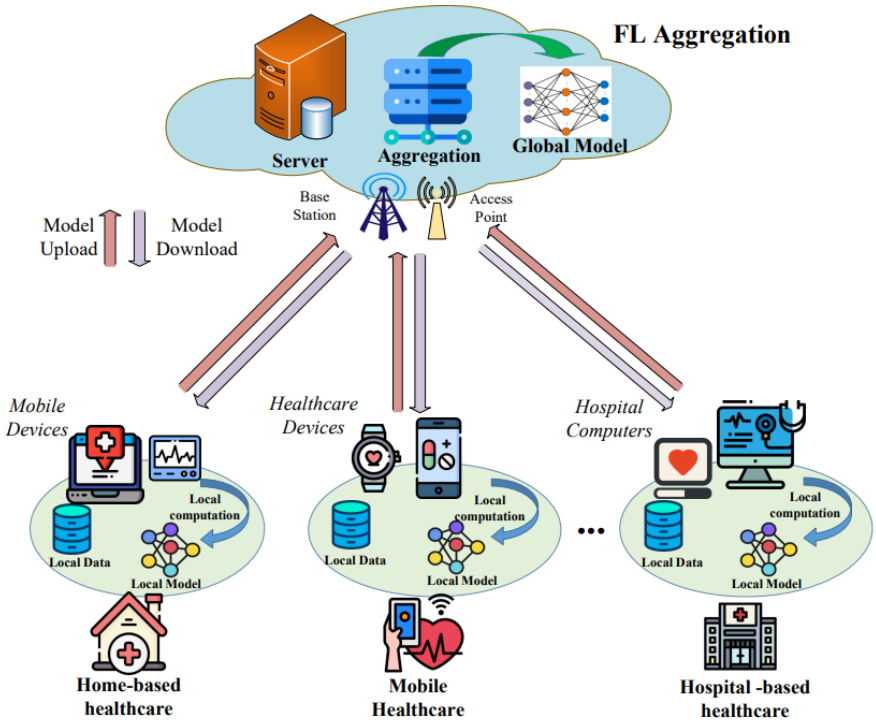


Figure 3.1 – Vue d'ensemble de l'architecture FL-Iomt

que la pression artérielle.

— Le serveur également connecté à Internet et à la passerelle de l'hôpital, il est responsable de la gestion des messages transmis par le dispositif mobile ainsi que de la transmission des messages aux dispositifs mobiles des patients.

Cette organisation garantit une évolutivité,car l'ajout de nouveaux dispositifs mobiles dans le réseau de l'hôpital augmente le trafic des messages et la logique du serveur de manière linéaire [56].

3.3 Algorithme FL-Iomt

L'apprentissage fédéré est un algorithme d'apprentissage automatique distribué qui construit un modèle global en faisant la moyenne des poids w sur de nombreux appareils au cours de plusieurs cycles de communication t.

Initialisation du système et sélection des clients

Le serveur d'agrégation sélectionne une tâche analytique en santé, telle que l'imagerie médicale automatique ou la détection des mouvements humains, ainsi que les exigences du modèle, telles que la classification de la tâche ou la prédiction de la tâche, et les paramètres d'apprentissage, tels que le nombre de nœuds neuronaux et les taux d'apprentissage. De plus, le serveur sélectionne un sous-ensemble de clients qui devraient participer au processus de FL.

Entraînement local distribué et mis à jour

Une fois le sous-ensemble des clients d'apprentissage déterminé, le serveur envoie un modèle initial comprenant un gradient global initial aux clients pour déclencher l'entraînement distribué. À chaque tour de communication, chaque client entraîne un modèle local en utilisant son propre ensemble de données et calcule sa mise à jour de modèle, par exemple le gradient dans les réseaux neuronaux. Ensuite, chaque client envoie sa mise à jour de modèle vers le serveur pour agrégation.

Agrégation du modèle et téléchargement

Après avoir reçu toutes les mises à jour des clients sélectionnés, le serveur les agrège en utilisant une méthode d'agrégation. Par exemple, nous pouvons utiliser l'approche de moyenne de modèle dans l'algorithme Federated Averaging (FedAvg) proposé par Google, où les paramètres de gradient des modèles locaux sont moyennés élément par élément avec des poids proportionnels à la taille des ensembles de données des clients. Ensuite, le serveur calcule une nouvelle version du modèle global et la diffuse à tous les clients comme base pour de futures mises à jour de modèles locaux lors du prochain tour d'apprentissage. Le processus de FL est répété jusqu'à ce que la fonction de perte globale converge ou que la précision souhaitée soit atteinte.

L'architecture du FL en santé comprend l'initialisation du système et la sélection des

clients, l'entraînement local distribué et les mises à jour des modèles, ainsi que l'agrégation du modèle et le téléchargement. Ce processus itératif permet d'obtenir un modèle global précis tout en préservant la confidentialité des données des clients.

3.4 quoi faire au cas de detections d'attaques

Les attaques considérées dans notre travail de recherche sont :

— Attaques de déni de service (DoS) : saturant le réseau ou le système cible.

— Attaques de déni de service distribué (DDoS) : provenant de multiples sources.

— Attaques de reconnaissance : collectant des informations sur le réseau cible.

— Attaques par force brute : tentatives de deviner des mots de passe.

— Attaques d'injection : injection de données malveillantes dans les requêtes.

— Attaques de buffer overflow : submersion de programmes avec des entrées excessives.

— Attaques de compromission de session : usurpation de sessions valides.

— Attaques de malware : infections par des logiciels malveillants.

— Attaques de contournement : tentative de contourner les mécanismes de sécurité.

Lorsque les IDS détectent une attaque dans le contexte des dispositifs médicaux et de la santé, les médecins et les professionnels de la santé peuvent prendre plusieurs mesures pour faire face à la situation comme :

— **Isoler le dispositif affecté** : Si l'attaque est détectée sur un dispositif médical spécifique, les médecins peuvent prendre des mesures pour isoler ce dispositif du reste du réseau. Cela peut aider à prévenir la propagation de l'attaque à d'autres dispositifs et à protéger les données et les fonctions critiques.

— **Arrêter le dispositif** : Dans certains cas, les médecins peuvent décider d'arrêter temporairement le dispositif affecté pour empêcher toute action malveillante ou

perturbation de ses opérations. Cela peut être crucial pour éviter d'aggraver la situation.

— **Analyser les données d'attaque** : Les médecins et les experts en sécurité peuvent examiner les données enregistrées par l'IDS pour comprendre la nature de l'attaque, son origine et son impact potentiel sur le dispositif ou le réseau. Cela peut aider à déterminer les mesures à prendre pour prévenir de futures attaques similaires.

— **Contacter les experts en sécurité informatique** : En cas d'attaque sérieuse ou complexe, les médecins peuvent collaborer avec des experts en sécurité informatique pour analyser en profondeur l'attaque, identifier les vulnérabilités exploitées et élaborer des stratégies de protection.

— **Appliquer des correctifs de sécurité** : Si l'attaque exploite une vulnérabilité connue, les médecins peuvent travailler avec les fournisseurs de dispositifs médicaux pour appliquer rapidement des correctifs de sécurité ou des mises à jour afin de prévenir des attaques similaires à l'avenir.

— **Rapporter l'incident** : Les médecins et les professionnels de la santé peuvent signaler l'incident aux autorités compétentes, aux agences de réglementation et aux organismes de sécurité informatique appropriés. Cela peut contribuer à renforcer les mesures de sécurité globales et à partager les leçons apprises pour améliorer la protection des dispositifs médicaux.

3.5 Repartition de données dans l'apprentissage fédéré

La répartition des données est l'un des aspects clés de l'apprentissage fédéré (FL), ca permet aux modèles d'apprentissage d'être entrainé avec des données décentralisées sans que les données ne quittent les dispositifs. La répartition des données est faite comme suite :

— **Les données :** Les données sont maintenues localement sur chaque appareil ou serveur participant.

— **Échantillonnage des Données :** Les données locales sont fractionnées en ensembles de données plus petits et cela est fait de facon Horizontal.

— **Fractionnement des Clients :** Lors de chaque cycle de communication, un ensemble de clients est sélectionné pour participer à la mise à jour du modèle, et cela se fait de facon aleatoire, puis une strategie de d'apprentissage fédéré sera appliqué sur eux.

3.6 Description du jeu de données UNSW-NB15

Dans cette section, nous allons parler de la configuration de l'environnement synthétique et la génération des détails de l'UNSW-NB15 [6].

3.6.1 Configuration du dataset

UNSWNB15 est créé en établissant l'environnement synthétique au laboratoire de cybersécurité de l'UNSW [6]. L'outil clé IXIA utilisé a permis de générer un représentant moderne du trafic réseau normal et anormal dans l'environnement synthétique (comme indique la figure 3.2).

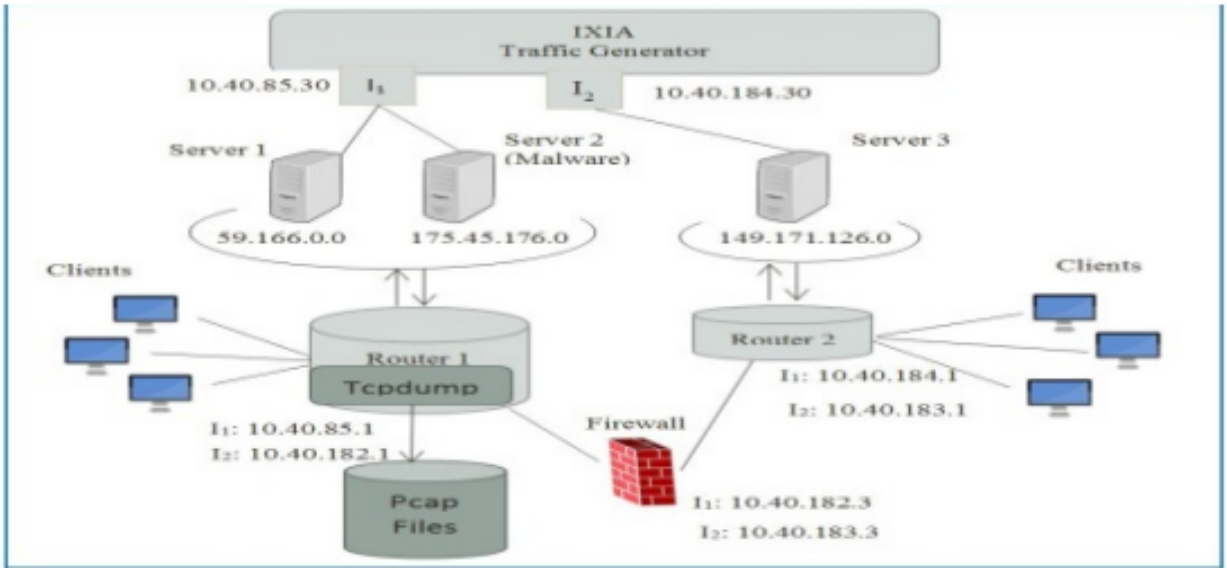


Figure 3.2 – la Visualisation Testbed de UNSW-NB15 [6]

Le jeu de données UNSW-NB15 [6] peut être utilisé pour l'apprentissage fédéré en santé de différentes manières. comme pour détecter le trafic malveillant dans les réseaux de santé. Le modèle pourrait être entraîné sur des données provenant de différents clients de l'Iomt, sans avoir besoin de partager les données brutes entre les entités du système de l'Iomt. Cela contribuerait à protéger la confidentialité des données des patients ce qui fait de lui une ressource précieuse pour la recherche en détection d'intrusion car il est bien documenté, vaste, complet, facile à utiliser et il couvre un large éventail de types d'attaques. [6].

3.7 Logiciels et librairies utilisés dans l'implémentation

Voici quelques frameworks d'apprentissage fédéré les plus populaires :

— **TensorFlow Federated** : TensorFlow Federated est un framework développé par

Google qui est utilisé pour implémenter des algorithmes d'apprentissage fédéré. TensorFlow Federated est open-source et est disponible pour Python et Java.

— **PySyft** : PySyft est un framework développé par OpenMined qui est utilisé pour implémenter des algorithmes d'apprentissage fédéré. PySyft est open-source et est disponible pour Python.

— **FATE** : FATE est un framework développé par WeBank qui est utilisé pour implémenter des algorithmes d'apprentissage fédéré. FATE est open-source et est disponible pour Python.

— **FLOWER** : Flower offre un langage stable et une implémentation ML Framework agnostic des composants core d'un système FL, Il procure une abstraction haut niveau pour les chercheurs qui peuvent expérimenter et implémenter de nouvelles idées. Flower permets d'avoir une transition rapide d'entraînement ML a des pipelines aux FL setup, pour évaluer la convergences des propriétés, et le temps d'entraînements. Et le plus importants c'est que Flower support les extensions des implémentation du [57].

Ce ne sont là que quelques nombreux frameworks d'apprentissage fédéré disponibles. Le meilleur framework pour une application particulière dépendra des exigences spécifiques de l'application. alors pour notre cas nous avant decider d'utiliser FLOWER.

3.7.1 Advantages d'utiliser un framework

Il existe plusieurs avantages à utiliser un framework plutôt que de développer un framework à partir de zéro en apprentissage fédéré. Ces avantages comprennent :

— **Gain de temps** : Les frameworks permettent de gagner beaucoup de temps en fournissant une base de code prête à l'emploi qui peut être utilisée pour implémenter des algorithmes d'apprentissage fédéré. Cela peut être particulièrement utile pour les chercheurs qui sont nouveaux dans le domaine de l'apprentissage fédéré ou qui n'ont pas le temps ou les ressources pour développer leur propre framework.

— **Réduction de la complexité** : Les frameworks peuvent également contribuer à réduire la complexité de l'apprentissage fédéré en fournissant une abstraction de haut niveau des algorithmes sous-jacents. Cela peut faciliter la compréhension et la mise en œuvre des algorithmes d'apprentissage fédéré, ainsi que le déploiement et la mise à l'échelle des applications d'apprentissage fédéré.

— **Amélioration de la sécurité** : Les frameworks peuvent également contribuer à améliorer la sécurité des applications d'apprentissage fédéré en fournissant des fonctionnalités de sécurité intégrées. Par exemple, certains frameworks offrent le chiffrement des données partagées lors de l'apprentissage fédéré, ainsi que des mécanismes permettant de s'assurer que les données ne sont pas altérées.

3.7.2 TensorFlow

Dans notre approache nous utilisent TensorFlow qui est une biblIothèque logicielle open source pour le calcul numérique de haute performance. Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (CPUs, GPUs, TPUs), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles. Initialement développé par des chercheurs et des ingénieurs de l'équipe de Google Brain au sein de l'organisation de l'IA de Google, il s'appuie sur l'apprentissage automatique et l'apprentissage en profondeur [58].

3.7.3 Google colabratory

Notre solution s'appuie sur Google Colab, un produit émanant de Google Research. Colab se présente comme un cahier de notes en ligne, totalement gratuit, offrant la possibilité à chacun d'écrire et de lancer du code Python directement depuis son navigateur. Cet environnement intégré se distingue par sa vocation à faciliter l'apprentissage automatique, l'analyse de données et l'éducation [59].

3.8 L'évaluation des performances d'un modèle

L'évaluation des performances d'un modèle dans le contexte de l'apprentissage fédéré est cruciale pour mesurer son efficacité et sa capacité à traiter les tâches spécifiques. Plusieurs méthodes peuvent être utilisées pour évaluer les performances d'un modèle, les approches courantes consiste à mesurer la précision et le rappel du modèle et La matrice de confusion et le score :

- **TP (True Positives)** vrais positifs est le nombre d'instances positives correctement classifiées
- **TN (True Negatives)** vrais négatifs est le nombre d'instances négatives correctement classifiées.
- **FP (False Positives)** faux positifs est le nombre d'instances négatives et qui sont prédites comme positives.
- **FN (False Negatives)** faux négatifs est le nombre d'instances positives classifiées comme négatives.
- **Precision** Il s'agit de la mesure des cas positifs correctement identifiés parmi tous les cas positifs prédits. Ainsi, il est utile lorsque le coût des faux positifs est élevé[60].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Rappel** C'est la mesure des cas positifs correctement identifiés parmi tous les cas positifs réels. C'est important lorsque le coût des faux négatifs est élevé [60].

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Accuracy** L'une des métriques les plus évidentes, c'est la mesure de tous les cas correctement identifiés. Il est surtout utilisé lorsque toutes les classes sont d'égale importance [61].

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

— **f1-Score** C'est la moyenne pondérée de l'accuracy et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. F1 est généralement plus utile que l'accuracy, surtout si vous avez une distribution de classe inégale. l'accuracy fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire [62].

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

3.9 Implementation de l'apprentissage fedéré

3.9.1 Importation des biblIothèques

— D'abord nous avons mis notre dataset dans google drive et nous avons monter google drive sur google collab pour acceder aux fichiers stockées.

— Ensuite nous avons installer la biblIotheque Flower qui facilite l'implementation et la simulations des algorithmes fédéré

— Puis nous avons installer les biblIotheques necessaire comme numpy, pandas, tensorflow, etc...

3.9.2 Préparation et Transformation des Données

Dans cette sous section, nous allons décrire les étapes essentielles de préparation et de transformation des données en vue de leur utilisation dans la construction et l'entraînement de modèles d'apprentissage automatique pour la détection d'intrusions.

3.9.3 Préparation des Données

La préparation des données est une étape cruciale dans tout processus d'apprentissage automatique. Elle vise à organiser, nettoyer et transformer les données brutes en un format adapté pour l'analyse et la modélisation. Dans cette section, nous détaillons les différentes étapes de préparation des données appliquées au jeu de données UNSW-NB15.

— Chargement des Données

Le jeu de données UNSW-NB15 [6] est divisé en deux parties : une partie pour l'entraînement et une partie pour le test. Les données sont chargées et stockées dans deux dataframes distincts : "dataset" et "dataset2".

— Nettoyage des Données

Les lignes contenant des valeurs nulles (NaN) sont supprimées des dataframes "dataset" et "dataset2". Cette opération garantit que seules les données complète sont conservées pour l'analyse ultérieure.

— Concaténation des Dataframes

Les dataframes "dataset" et "dataset2" sont concaténés pour former un seul dataframe consolidé appelé "datasetf". Cette étape facilite la manipulation et la transformation ultérieures des données.

— Séparation des Caractéristiques Catégoriques et Numériques

Les caractéristiques du jeu de données sont divisées en deux groupes : les caractéristiques catégoriques et les caractéristiques numériques.

— Encodage One-Hot des Caractéristiques Catégoriques

Les caractéristiques catégoriques sont soumises à un encodage one-hot pour les préparer à être utilisées dans les modèles d'apprentissage automatique. Cet encodage permet de représenter les catégories sous forme de vecteurs binaires.

3.9.4 Normalisation des Caractéristiques Numériques

La normalisation des caractéristiques numériques est une étape cruciale visant à mettre toutes les variables sur une échelle cohérente. Ceci garantit que les caractéristiques numériques n'exercent pas un effet disproportionné sur les modèles d'apprentissage. Cette sous section expose en détail les étapes de normalisation appliquées aux caractéristiques numériques du jeu de données.

— Préparation des Caractéristiques Numériques

La préparation des caractéristiques numériques est accomplie par le biais d'un MinMaxScaler, transformant ainsi les caractéristiques numériques en une plage normalisée, entre -1 et 1.

— Concaténation des Caractéristiques Numériques et Catégoriques

À cette étape consiste à combine harmonieusement les caractéristiques numériques normalisées et les caractéristiques catégoriques préalablement encodées en vue d'une utilisation efficace dans les modèles d'apprentissage automatique.

3.10 Architecture du Modèle de Réseau de Neurones

L'architecture du modèle suit une structure séquentielle, qui est une série de couches connectées. Chaque couche contient des neurones qui traitent les données d'entrée et les transforment en informations significatives. Dans le contexte de ce scénario d'apprentissage fédéré de l'Iomt, le réseau de neurones vise à effectuer des tâches de classification binaire, telles que la détection de certaines conditions médicales ou anomalies. Voici un aperçu de l'architecture :

— **Couche d'Entrée**: La couche d'entrée est initiée avec 80 neurones, correspondant à la dimensionnalité des données d'entrée. Cette couche sert de point d'entrée pour les données collectées à partir des dispositifs IoMT.

— **Couches Cachées**: Après la couche d'entrée, il y a plusieurs couches cachées. Chaque couche cachée comprend un nombre variable de neurones, et chaque neurone utilise une fonction d'activation linéaire rectifiée (ReLU). Cette fonction d'activation introduit la non-linéarité, permettant au modèle de capturer des relations complexes au sein des données.

  — La première couche cachée est composée de 40 neurones.

  — La deuxième couche cachée a 30 neurones.

  — La troisième couche cachée contient 20 neurones.

  — La quatrième couche cachée incorpore 10 neurones.

— **Couche de Sortie**: La dernière couche du réseau est constituée d'un seul neurone. Elle utilise la fonction d'activation sigmoïde, qui produit une valeur entre 0 et 1. Cette valeur représente la probabilité prédite du résultat de la classification binaire.

L'architecture que nous avons conçue est d'une importance capitale pour notre système de détection d'intrusions par anomalies basé sur l'apprentissage fédéré. L'utilisation d'un modèle de deep learning s'est avérée indispensable pour garantir la confidentialité des données tout en profitant de la puissante capacité de généralisation propre à ces modèles. Cette approche est privilégiée pour sa capacité à décrypter les structures complexes et les schémas à partir des données, ce qui permet une détection précise des anomalies. Ceci est particulièrement vital dans le domaine de la santé où la précision est primordiale.

Dans cette perspective, l'ajout de l'apprentissage supervisé à notre système renforce sa résistance et sa capacité à identifier les modèles d'anomalies dans les données. En fournissant au modèle des exemples préalablement étiquetés, l'apprentissage supervisé lui permet de développer une connaissance approfondie des caractéristiques spécifiques aux anomalies médicales. Cette méthodologie bien pensée, basée sur des données pré-étiquetées, confère à notre solution un avantage majeur dans la détection proactive et précise des intrusions, tout en préservant la confidentialité des données sensibles propres à l'IoMT.

3.11 Hyperparamètre de l'apprentissage fédéré

Une fois la topologie choisie, nous pouvons contrôler différents paramètres du processus d'apprentissage fédéré afin de l'optimiser par opposition à celui déjà utilisé par le modèle:

— Nombre de cycles d'apprentissage fédéré.

— Nombre total de clients utilisés dans le processus.

— Fraction d'instance client utilisée pour chaque itération.

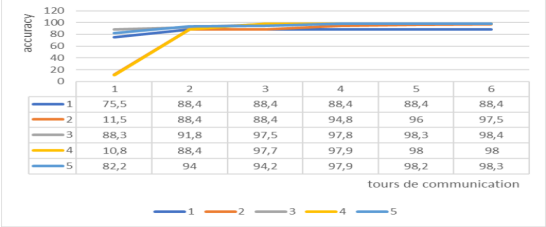
— Nombre d'epoch utilisé à chaque itération d'apprentissage.

3.12 Adaptation des Hyperparamètres et Simulation de l'evaluations d'apprentissage avec les différentes stratégies

Dans cette section, nous détaillons la simulation des évaluations d'apprentissage avec différentes stratégies dans le contexte de l'apprentissage fédéré. Nous avons défini une fonction d'évaluation pour le modèle, qui sera utilisée pour évaluer la performance après chaque cycle d'apprentissage sur le serveur central. Le modèle a été compilé en spécifiant la fonction de perte, l'optimiseur et les métriques à utiliser pour l'évaluation.

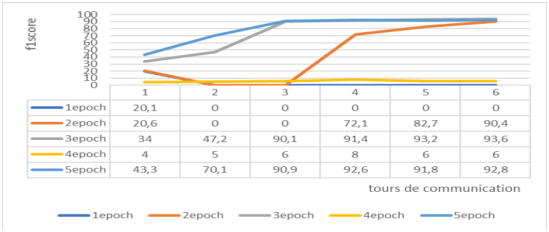
Nombre d'epoch

Afin de savoir qu'elle est le meilleur nombre d'epoch à choisir pour avoir de bons résultats, nous avons fixé le nombre de clients à une valeurs égale à 100 et le nombre minimum de clients que le serveur doit sélectionner pour effectuer l'entraînement lors d'une ronde d'apprentissage fédéré a une valeur égale à 5 et augmenter le nombre d'epoch à chaque fois.



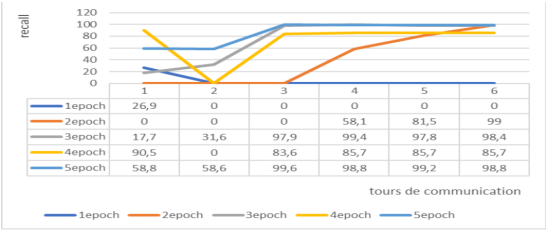
	1	2	3	4	5	6
1	75,5	88,4	88,4	88,4	88,4	88,4
2	11,5	88,4	88,4	94,8	96	97,5
3	88,3	91,8	97,5	97,8	98,3	98,4
4	10,8	88,4	97,7	97,9	98	98
5	82,2	94	94,2	97,9	98,2	98,3

Figure 3.3 – accuracy / tours de communication



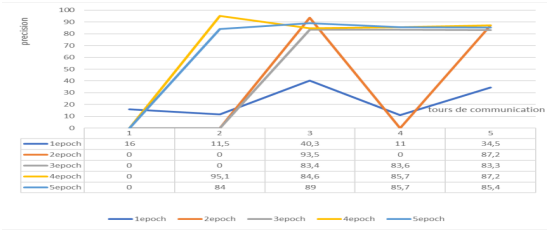
	1	2	3	4	5	6
1epoch	20,1	0	0	0	0	0
2epoch	20,6	0	0	72,1	82,7	90,4
3epoch	34	47,2	90,1	91,4	93,2	93,6
4epoch	4	5	6	8	6	6
5epoch	43,3	70,1	90,9	92,6	91,8	92,8

Figure 3.4 – f1-score / tours de communication



	1	2	3	4	5	6
1epoch	26,9	0	0	0	0	0
2epoch	0	0	0	58,1	81,5	99
3epoch	17,7	31,6	97,9	99,4	97,8	98,4
4epoch	90,5	0	83,6	85,7	85,7	85,7
5epoch	58,8	58,6	99,6	98,8	99,2	98,8

Figure 3.5 – recall / tours de communication



	1	2	3	4	5
1epoch	16	11,5	40,3	11	34,5
2epoch	0	0	93,5	0	87,2
3epoch	0	0	83,4	83,6	83,3
4epoch	0	95,1	84,6	85,7	87,2
5epoch	0	84	89	85,7	85,4

Figure 3.6 – precision / tours de communication

Figure 3.7 – representation du changement de l'accuracy, recall, f1-score, precision avec nombre d'epoch fonction du nombre de tours de communication

Les figures 3.3, 3.4, 3.5, 3.6 et 3.12 présentes les résultats du processus d'apprentissage fédéré à différents tours, après l'ajout d'epochs supplémentaires. Les courbes tracent les variations de l'accuracy, du recall, du f1-score et de la précision en fonction du nombre de tours de communication. Une observation significative émerge : en augmentant le nombre d'epochs, l'accuracy démontre une amélioration notable.

Dans la Figure 3.3, les courbes (1) et (5) se démarquent. La courbe (1) affiche une amélioration modeste, atteignant une précision maximale de 88,4%, tandis que la courbe (5) se distingue par une convergence rapide autour du troisième tour, obtenant une précision maximale de 98,3%.

La Figure 3.5 révèle que la courbe (1) a une performance médiocre dès le deuxième tour, tandis que la courbe (5) affiche des résultats bien meilleurs, atteignant 92,8% et 98,8% de précision après cinq tours respectivement.

En ce qui concerne la Figure 3.4, les courbes (1) et (2) se caractérisent par leur insta-

bilité et leur faible performance, en contraste avec la courbe (5) qui se révèle nettement plus performante.

En somme, l'analyse des résultats indique que l'augmentation du nombre d'epochs influe positivement sur l'accuracy. Les courbes révèlent des trajectoires variables en fonction des tours, avec des convergences rapides et des performances maximales qui diffèrent selon les courbes et les figures.

résultats

Après l'analyse des résultats et de l'étude des métriques de performance, nous avons constaté qu'augmenter le nombre d'epoch permet de réduire le nombre de tours nécessaire pour converger. Autrement dit plus on augmente le nombre d'epoch plus le graphe atteint son seuil et converge rapidement à un certain nombre d'epoch.

Nombre minimum de clients que le serveur doit sélectionner pour effectuer l'entraînement

D'après l'expérience sur le nombre d'epoches et ses effets sur le nombre de tours de communication, nous avons constaté qu'augmenter le nombre d'epoch permet d'avoir de meilleurs résultats, alors pour savoir quel est le nombre de minimum de clients optimale pour avoir un bon résultat de convergence nous avons fixé le nombre d'epoch a 5 et le nombre de clients à 100 et augmenter le nombre minimal de client sélectionnées a chaque tours.

Les graphiques 3.8, 3.9, 3.10 et 3.11 présentent les courbes illustrant l'évolution de l'accuracy, de la précision, du rappel (recall), du score F1 et du nombre minimum de clients sélectionnés en fonction du nombre de tours de communication. Une observation majeure ressort : le niveau de convergence associé à un nombre minimum de clients sélectionnés de 45 est inférieur à celui obtenu avec un nombre minimum de clients sélectionnés de 5. De plus, il est démontré que la réduction du nombre minimum de clients sélectionnés conduit à de meilleurs résultats de convergence.

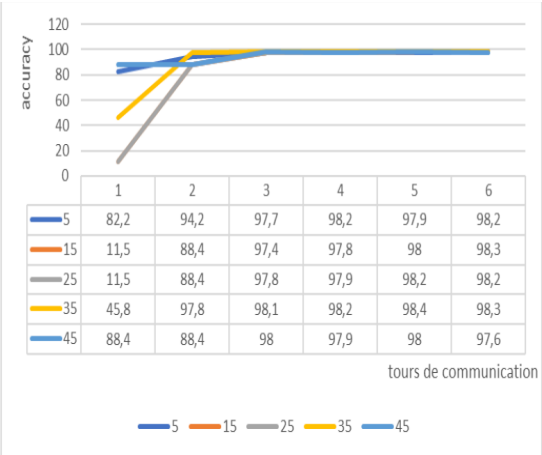


Figure 3.8 – accuracy avec minfitclient / tours de communication

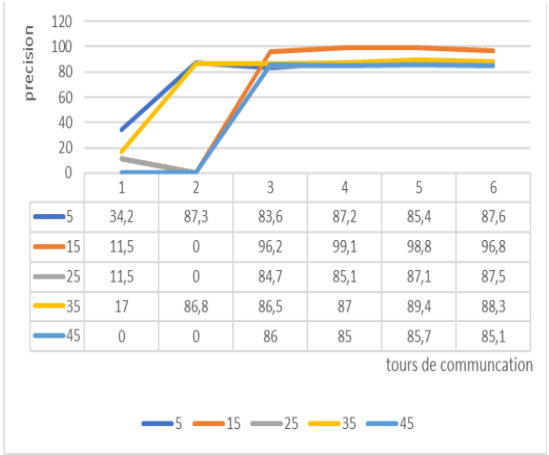


Figure 3.9 – f1-score avec minfitclient / tours de communication

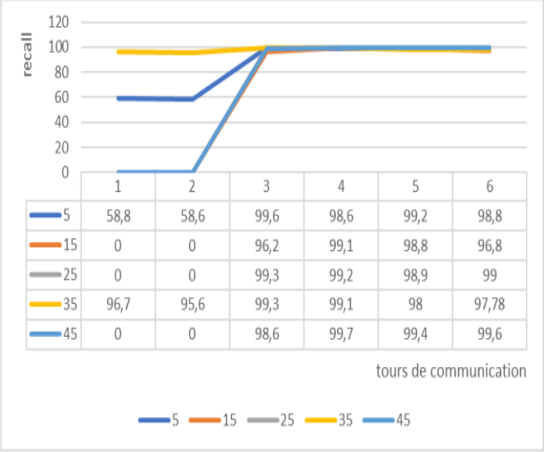


Figure 3.10 – recall avec minfitclient / tours de communication

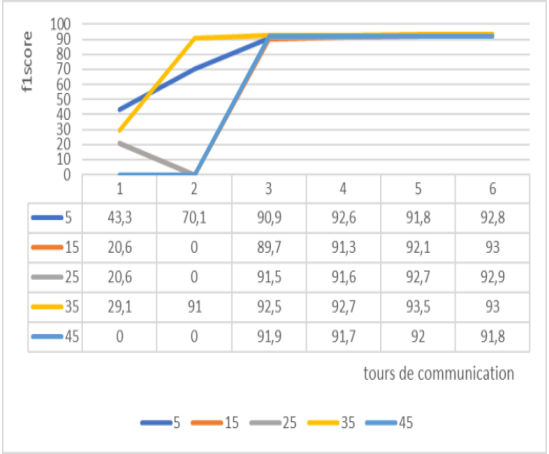


Figure 3.11 – precision minfitclient / tours de communication

Figure 3.12 – representation du changement de l'accuracy, recall, f1-score, precision avec nombre minimum de clients selectionées en fonction du nombre de tours de communication

Dans la Figure 3.8, la courbe 2 (nombre minimum de clients sélectionnés = 15) affiche des résultats de convergence prometteurs. Bien qu'elle atteigne une valeur de 98.3% dès le troisième round, elle nécessite plus de tours pour ce faire par rapport à la courbe 1. En revanche, dans la Figure 3.10, la plupart des courbes convergent vers leurs seuils respectifs autour du cinquième tour, même si elles commencent à converger dès le troisième tour.

En observant la Figure 3.11, on constate que les courbes sont relativement similaires dès le troisième tour, affichant des taux de convergence oscillant entre 90% et 93%. Cela suggère que des variations subtiles dans le nombre minimum de clients sélectionnés peuvent avoir un impact relativement limité sur la convergence à ce stade.

En synthèse, l'analyse des résultats révèle que réduire le nombre minimum de clients sélectionnés améliore la convergence et que, dans certains cas, une diminution significative de ce nombre peut conduire à une convergence plus rapide sans sacrifier la qualité des résultats.

résultats

D'après le graphe qui illustre la courbe du changement du recall, f1-score, et tous les autres métriques au fil des tours de communication, on remarque que plus le nombre minimum de clients sélectionnées est inférieur, plus les résultats de convergence sont meilleurs.

Nombre de clients total

Après les deux expériences, on a voulu savoir quel nombre de clients serait-il le plus optimal pour réduire le nombre de tours de communication donc on a : fixer les nombres minimum de clients sélectionnées à 5 et le nombre d'epoch à 5 et augmenter le nombre de clients total à chaque fois.

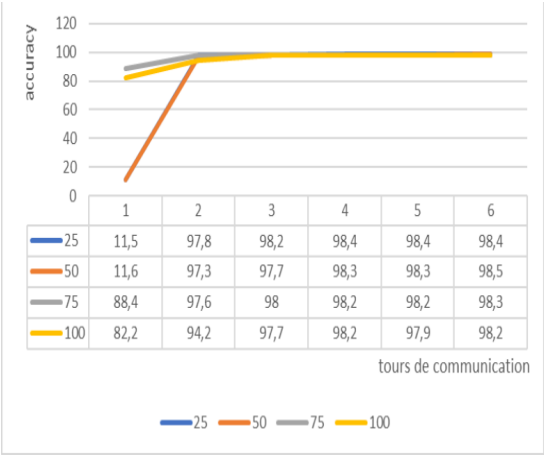


Figure 3.13 – accuracy / tours de communication

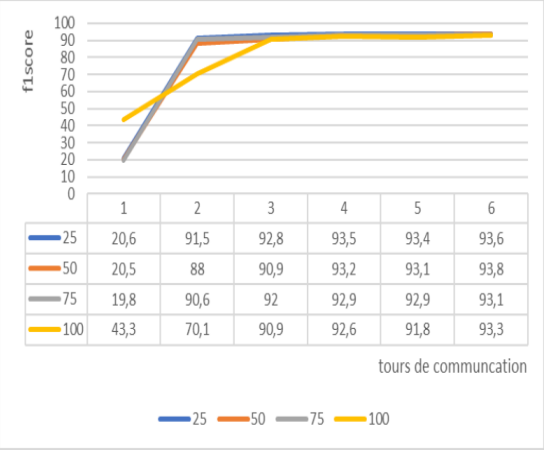


Figure 3.14 – f1-score / tours de communication

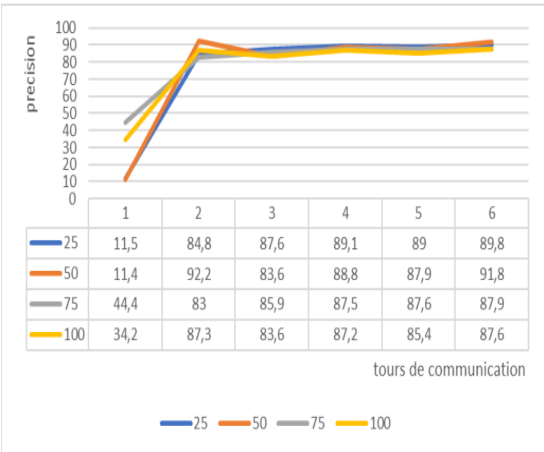


Figure 3.15 – precision / tours de communication

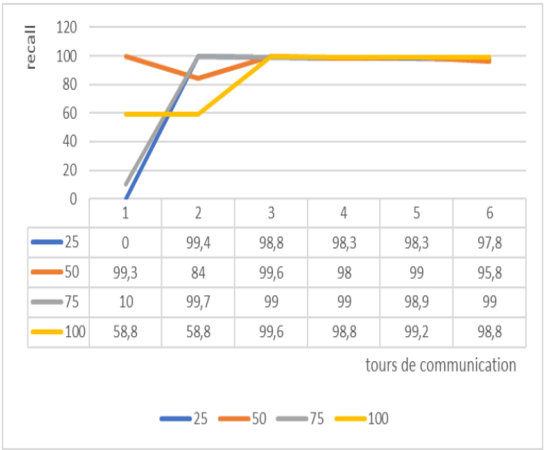


Figure 3.16 – recall minfitclient / tours de communication

Figure 3.17 – L'effet de l'augmentation de nombre clients sur l'optimisation des tours de communication sur le recall, accuracy f1-score, precision

résultats

Dans la Figure 3.13 , nous observons que l'augmentation du nombre de clients n'affecte pas significativement le taux de convergence des métriques. Les courbes atteignent rapidement un niveau maximal de convergence qui est egale a 98% en terme d'accuracy et 93% en f1score , ce qui reflète que l'ajout de plus de clients n'entraîne pas d'amélioration remarquable de la convergence des métriques de performances, tous comme la Figure 3.14.

En examinant la précision des résultats dans le contexte de différents nombres de clients lorsque le nombre de clients est égal à 100 clients, la courbe de précision affiche un saut remarquable dès le deuxième tour de communication. Cette courbe se stabilise ensuite à une valeur optimale, presque égale aux valeurs obtenues pour d'autres nombres de clients, soit environ 88%. Ces résultats des figures 3.17 démontre que l'ajout de plus de clients n'a pas d'impact significatif sur la précision du modèle, et que cette précision converge rapidement vers un niveau optimal dès les deuxièmes tours de communication.

Comparaison entre les differentes methodes de l'apprentissage fédéré

Afin de comparer entre les deux approches federer et l'approche centralisé, nous avons testé les 2 méthode non-adaptive (FedAvg, FedAvgM) et adaptive (FedYogi, FedAdagrad, FedAdam) de l'approche federer afin d'étudier leurs performance en fonction d'accuracy, f1-score, la matrice de confusion et de tours de communication, avec des paramètre égal un nombre de clients = 100 et un nombre minimum de clients selectionées = 5 et un nombre d'epoch égal à 5.

la figure 3.18 démontre que la méthode FEDAVGM et FEDAVG et FEDYOGI Obtiennent un bon résultat qui augmente au fil des deux premiers tours et stagne au bout du troisième tour pour se fixer a sa valeur maximale, et tout le contraire pour les méthodes FEDADAGRAD et FEDADAM qui ont une courbe de f1-score faible et instable.

les graphes 3.18 démontrent les résultats des différentes méthodes adaptive et non adaptive de l'apprentissage fédérer en termes de tours de communications et de différentes

accuracy	1	2	3	4	5	6
FEDAVG	82,2	94,2	97,7	98,2	97,9	98,2
FEDAVGM	88,4	94,6	97,7	97,8	97,9	98,1
FEDADAGRAD	11,2	88,4	88,4	88,4	11,5	88,4
FEDYOGI	11,5	67,3	94,6	95	95,4	96,7
FEDADAM	88,4	88,4	88,4	11,3	11,5	88,4

f1score	1	2	3	4	5	6
FEDAVG	43	70,1	90,9	91,8	92,8	93,2
FEDAVGM	0	71	90,9	91,5	91,6	92,5
FEDADAGRAD	18,2	0	0	0	20,6	0
FEDYOGI	20,6	39,6	74,4	73,5	73,7	73,7
FEDADAM	0	0	0	20,6	20,6	0

precision	1	2	3	4	5	6
FEDAVG	34,2	87,3	83,6	87,2	85,4	87,6
FEDAVGM	0	92	83,6	84,9	85	86,8
FEDADAGRAD	10,2	0	0	0	11,51	0
FEDYOGI	11,51	25,2	82,1	87,3	94,4	97,6
FEDADAM	0	0	0	11,51	11,51	0

recall	1	2	3	4	5	6
FEDAVG	58,8	58,6	99,6	98,8	99,2	98,8
FEDAVGM	0	58,6	99,6	99,3	99,4	98,9
FEDADAGRAD	86	0	0	0	1	0
FEDYOGI	0	93,1	68,1	63,5	60,4	63,3
FEDADAM	0	0	0	1	1	0

Figure 3.18 – la différence entre les strategies en fonction de tours de communication avec recall, accuracy, f1-score, precision

métriques de performances.

résultats

Les approches centralisées ont affiché de bons résultats en termes de performances, mais elles posent un problème de confidentialité en exigeant la collecte de données auprès d'une seule source. Cela crée une vulnérabilité puisqu'un attaquant pourrait cibler cette source unique pour accéder à toutes les données.

Parmi les méthodes, les non adaptatives ont montré des performances solides avec seulement 2 cycles de communication pour converger vers une précision maximale de 98% en termes d'exactitude, 93% et 92% respectivement. En comparaison, les méthodes adaptatives (fedadam, fedyogi, fedadagrad) ont obtenu des résultats inférieurs aux non adaptatives. Cependant, certaines d'entre elles ont présenté des avantages en termes de précision. Par exemple, la méthode fedYogi a atteint 97%, se rapprochant ainsi du résultat des méthodes non adaptatives (96.7%). En revanche, les autres méthodes adaptatives ont montré des résultats instables et peu fiables.

3.13 Conclusion

En synthèse, cette section dédiée à la conception expose notre système dédié pour la détection des intrusions, fondé sur l'apprentissage fédéré assorti de diverses stratégies. Dans le cadre de cette approche, nous avons entrepris pour résoudre les problématiques au paramétrage du modèle, ainsi qu'à l'optimisation des performances, à savoir accuracy et la minimisation des cycles de communication. Subséquemment, nous avons procédé à une comparaison méthodique des différentes techniques de FL, en vue de sélectionner la plus performante et de la confronter à l'approche centralisée. et tenir en compte de l'étape évolutive de l'apprentissage fédéré et de son aptitude à minimiser le risque de perte de données tout en préservant la confidentialité de manière plus efficace que l'approche centralisée, les résultats obtenus laissent entrevoir une prochaine substitution du modèle

centralisé par l'apprentissage fédéré dans un futur proche.

Conclusion Générale

En synthèse, cette étude s'est concentrée sur l'intégration du ML afin de renforcer la sécurité et préserver la confidentialité au sein de l'Iomt, une composante spécifique de l'Iot. Les fondements du ML ainsi que du DL ont été minutieusement exposés pour établir les bases nécessaires. En parallèle, une exploration approfondie du FL a été entreprise, mettant en exergue son architecture et les diverses stratégies qu'il implique. À travers une analyse critique des solutions de sécurité basées sur le FL pour l'Iomt, une proposition novatrice de détection d'intrusion reposant sur le ML et le concept de FL a été formulée.

La solution proposée s'appuie sur le concept de FL pour détecter les intrusions au sein du système de l'Iomt. Notre étude a été orientée vers l'optimisation de la communication au sein du réseau Iomt afin de perfectionner la détection des intrusions. Pour ce faire, nous avons implémenté un modèle de ML au niveau de chaque client, puis procédé à des expérimentations poussées en variant les stratégies de communication et en ajustant plusieurs paramètres clés dans le processus de FL.

Une série de tests minutieux a été entreprise, incluant des variations du nombre d'epochs, du nombre de clients et des tours de communication pour chaque stratégie envisagée. Les métriques de performance, telles que l'accuracy, la précision, le rappel, le score F1 ont été évaluées et rigoureusement comparées. Ces évaluations ont été menées en utilisant le jeu de données UNSW-NB15.

En vue de valider notre solution, une comparaison rigoureuse avec la méthode centralisée a été menée. Les résultats obtenus révèlent que le modèle FL optimisé se rapproche des performances du modèle centralisé tout en offrant un niveau de protection supplé-

mentaire pour la confidentialité des données. En somme, notre approche témoigne de la faisabilité d'une solution alliant efficacité, sécurité et préservation des données confidentielles au sein de l'Iomt.

Dans nos futurs travaux, nous envisageons d'étendre notre solution en entreprenant plusieurs initiatives clés. Tout d'abord, nous prévoyons de mettre à l'épreuve notre solution en la testant sur d'autres ensembles de données, afin de valider son efficacité et sa robustesse dans divers contextes. En parallèle, nous accorderons une attention particulière à la sécurité de la communication entre les modèles déployés chez les clients et le serveur central. Pour renforcer davantage la fiabilité de notre approche, nous mettrons en œuvre des mécanismes visant à contrer les attaques d'empoisonnement de données, assurant ainsi l'intégrité des informations partagées.

Bibliography

[1] Sabeen Tahir, Sheikh Bakhsh, Maysoon Abulkhair, and Madini Alassafi. An energy-efficient fog-to-cloud internet of medical things architecture. *International Journal of Distributed Sensor Networks*.

[2] Mesaiahmed Hamza and Nadirkais. *Commande de la machine asynchrone à double alimentation – apport des techniques de l'intelligence artificielle*. PhD thesis, Université Djilali Liabes de Sidi Bel Abbes, Faculté de génie électrique, Département d'Électrotechnique, 2017.

[3] Raycad. Un réseau de neurones. Medium, Novembre 2017.

[4] Dinh C. Nguyen, Quoc-Viet Pham, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *arXiv preprint arXiv:2111.08834*, 2021.

[5] S. Ruder. Un aperçu des algorithmes d'optimisation de descente de gradient. Website, Unknown.

[6] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015.

[7] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, Jun. 2015.

[8] Unknown. Objets médicaux connectés : les défis au-delà de la cybersécurité, Unknown.

[9] Cybersecurity-Ventures. Healthcare industry to spend $125 billion on cybersecurity from 2020 to 2025, 2021.

[10] J.P. Ezikola Mazoba. *L'étude de l'internet des objets et contrôle d'accès aux données*. PhD thesis, Université de Panafricaine, 2015.

[11] Jayavardhana Gubbi, Rajkumar Buyya, Srdjan Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[12] Santosh Madakam, R. Ramaswamy, and S. Tripathi. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(3):164–173, 2015.

[13] Argyro Mavrogiorgou and et al. Internet of medical things (iomt): Acquiring and transforming data into hl7 fhir through 5g network slicing. *Emerging Science Journal*, 3(2):64, 2019.

[14] L. Mineti, I. Patrono, and A. Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. In *Proc. SoftCOM*, Sept. 2011.

[15] IEEE 802 Working Group. Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans). Technical Report 4, IEEE Std 802, 2011.

[16] A.J. Jara, M.A. Zamora, and A.F. Skarmeta. Knowledge acquisition and management architecture for mobile and personal health environment based on the internet of things. In *Proc. Int. Conf. TrustCom*, pages 1811–1818, 2012.

[17] Nikhat Akhtar and Yusuf Perwej. The internet of nano things (iont) existing state and future prospects. *GSC Advanced Research and Reviews (GSCARR)*, 5(2):131–150, 2020.

[18] Unknown. Business logic attack, Unknown.

[19] H. Ahmadi, G. Arji, L. Shahmoradi, R. Safdari, M. Nilashi, and M. Alizadeh. The application of internet of things in healthcare: A systematic literature review and classification. *Univ. Access Inf. Soc.*, 18:837–869, 2019.

[20] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey of internet of things: architecture enabling technologies security and privacy and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, 2017.

[21] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman. Internet of medical things (iomt): applications benefits and future challenges in healthcare domain. *JCM*, 2017.

[22] T. Yaqoob, H. Abbas, and M. Atiquzzaman. Security vulnerabilities attacks countermeasures and regulations of networked medical devices—a review. *IEEE Commun. Surveys Tuts.*, 21(4):3723–3768, 4th Quart. 2019.

[23] Fortinet. Intrusion detection system. `https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system`, Accessed.

[24] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Unknown*.

[25] Ahmad Khraisat and Mamoun Alazab. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, 4:18, 2021.

[26] Pariwat Ongsulee. Artificial intelligence, machine learning and deep learning. In *2017 15th International Conference on ICT and Knowledge Engineering (ICTKE)*, pages 1–6, 2017.

[27] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, David Huba, Ariel Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konecny, Stefano Mazzocchi, H. Brendan McMahan, Thijs van Overveldt, Daniel Petrou, Daniel Ramage, and Jonas Roselander. Towards federated learning at scale: System design. *IEEE International Conference on Systems and Machine Learning*, 2019.

[28] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan Publishing, New York, 1994.

[29] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[30] Unknown. Neural network models and deep learning - a primer for biologists. *Unknown*, Unknown.

[31] IBM. Ai vs. machine learning vs. deep learning vs. neural networks, Unknown.

[32] IONOS by 11. Deep learning vs. machine learning, Unknown.

[33] Konstantinos Chatzikokolakis and et al. Federated learning: A survey. *arXiv preprint arXiv:1902.01046*, 2019.

[34] Hui Zhang and et al. Federated learning: Challenges, methods, and applications. *arXiv preprint arXiv:1901.05538*, 2019.

[35] Hui Zhang and et al. Federated learning for healthcare: A review. *arXiv preprint arXiv:2003.08809*, 2020.

[36] Ameet Talwalkar and et al. Federated learning: A new approach to machine learning on mobile devices. *arXiv preprint arXiv:1602.05629*, 2016.

[37] Raza Vig and et al. Federated wireless health: A novel framework for privacy-preserving machine learning in healthcare. *arXiv preprint arXiv:2003.10840*, 2020.

[38] Yuxiang Cheng, Yang Liu, Tianhui Chen, and Qiang Yang. Federated learning for privacy-preserving ai. *Communications of the ACM*, 63(12):33–36, 2020.

[39] S. Yang, B. Ren, X. Zhou, and L. Liu. Parallel distributed logistic regression for vertical federated learning without third-party coordinator. *arXiv preprint arXiv:1911.09824*, 2019.

[40] S. Sharma, C. Xing, Y. Liu, and Y. Kang. Secure and efficient federated transfer learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2569–2576, 2019.

[41] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An in-depth study of lte: effect of network protocol and application behavior on performance. *SIGCOMM Computer Communication Review*, 43(4):363–374, 2013.

[42] C. Van Berkel. Multi-core for mobile phones. In *Proc. Conf. Design, Automation and Test in Europe*, pages 1260–1265, 2009.

[43] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In *Proc. Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[44] J. Li, M. Khodak, S. Caldas, and A. Talwalkar. Differentially private meta-learning. In *Proc. Int. Conf. Learning Representations*, 2020.

[45] Vishwa Amitkumar Patel, Pronaya Bhattacharya, Sudeep Tanwar, and Rajesh Gupta. Adoption of federated learning for healthcare informatics: Emerging applications and future directions. *IEEE*, 2022.

[46] Mohamed Amine Ferrag, Othmane Friha, Leandros Maglaras, Helge Janicke, and Lei Shu. Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE*, 2021.

[47] Nour Moustafa. The bot-iot dataset, 2019.

[48] Ivan Vaccari, Giovanni Chiola, Maurizio Aiello, Maurizio Mongelli, and Enrico Cambiaso. Mqttset, a new dataset for machine learning techniques on mqtt. *Sensors*, 20, 11 2020.

[49] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. $\mathrm{Ton}_i ottelemetrydataset : Anewgenerationdatasetofiotandiiotfordata - drivenintrusiondetectionsystems. IEEE Access, 8 : 165130 - -165150, 2020.$

[50] Joseph Bamidele Awotunde, Kazeem Moïse Abiodun, Emmanuel Abidemi Adeniyi, Sakinat Oluwabukonla Folorunso, and Rasheed Gbenga Jimoh. A deep learning-based intrusion detection technique for a secured iomt system. In *Communications in Computer and Information Science book series (CCIS, volume 1547)*. Springer, 2022.

[51] Authors of the Book Chapter. *Title of the Book Chapter*, chapter Chapter Number. Springer, Year.

[52] Yazan Otoum, Yue Wan, and Amiya Nayak. Federated transfer learning-based ids for the internet of medical things (iomt). In *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021.

[53] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018.

[54] Mohammed Aledhari, Raja Razzak, Reza M. Parizi, and Fahad Saeed. Intelligent intrusion detection based on federated learning for edge-assisted internet of things. *Unknown*, 2020.

[55] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

[56] William Schneble and Geethapriya Thamilarasu. Attack detection using federated learning in medical cyber-physical systems. 2019.

[57] Daniel J. Beutel, Taner Topal, Akhil Mathur, Haoyu Qian, Lingjing Yang, Yang Zhang, Arunesh Pal, Deva Ramanan, and Andreas Krause. Flower: A friendly federated learning framework. *arXiv preprint arXiv:2007.14390*, 2020.

[58] TensorFlow Authors. Tensorflow. `https://www.tensorflow.org/`, Accessed.

[59] Google Research. Google colaboratory - frequently asked questions. `https://research.google.com/colaboratory/faq.htm`, Accessed.

[60] Unknown. Definitive guide to accuracy, precision, recall for product developers, Unknown.

[61] Unknown. Accuracy and loss, Unknown.

[62] Unknown. Confusion matrix, Unknown.