République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes distribués (R.S.D)

Thème

Système de Détection d'Intrusion basé sur l'apprentissage fédéré dans le Fog Computing

Réalisé par : Melle Benseddik Yousra & Melle Benmiloud Amina

Présenté le 25 Juin 2023 devant le jury composé de :

Mr MANA Mohammed	**Président**
Mr BAMBRIK Ilyas	**Examinateur**
Mme LABRAOUI Nabila	**Encadrante**
Mme BENSAID Rajaa	**Co-Encadrante**

Année universitaire : 2022-2023

Dédicaces

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon très cher père

Tu as toujours été à mes côtés pour me soutenir et m'encourager. Je te témoigne pour tous les efforts et les sacrifices que tu n'as jamais cessé de consentir pour mon instruction et mon bienêtre.

À mes bien-aimés parents, vous représentez les personnes les plus précieuses à mes yeux.

A mes très chers frères

"Houssem Eddine" et "Alaà Eddine" qui n'ont pas cessé de me conseiller, encourager et soutenir tout au long de mes études. Que Dieu les protège et leur offre la chance et le bonheur.

A ma chère famille

Mes neveux adorés "Anas" et "Yazan", et ma belle-sœur "Ghofrane". Que Dieu les protège et leur donne le bonheur.

Ma grand-mère, tantes et oncles, leurs époux et épouses, et mes cousines "Samah", "Kawtar", "Bouchra" et "Sarah". Que Dieu leur donne une longue et joyeuse vie.

A mes chères amies

Mon binôme et ma meilleure amie "Amina" pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Ma meilleure amie "Riham" qui nous a encouragés tout au long du projet.

"Yousra"

Dédicaces

Louange à dieu avant tout…

Avec tout mon amour éternel et avec l'intensité de mes émotions je dédie cet humble travail:

A ceux, qui sans eux rien n'aurait pu être…Mes parents. Rien ne peut exprimer l'amour, l'estime, le dévouement et le respect que j'ai pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Vous avez su m'inculquer les valeurs nobles de la vie, m'apprendre le sens du travail, et de la responsabilité. Je n'arriverai jamais à vous remercier autant que vous le méritez. Merci pour votre esprit attentif et le soutien que je trouve toujours auprès de vous. Que Dieu vous accord une longue vie pleine de santé.

A mes chers frères "Fares " "Tayeb" et "Mohamed", qui m'encouragent et m'aident tout au long de mes études, que Dieu le protège.

A mes adorables sœurs "Soumia" "Ahlem" et "Sarah" qui m'ont toujours soutenu, que Dieu les protège.

À toute ma chère famille mes tantes, mes oncles, mes cousins et cousines.

Merci à tous mes amis qui m'ont soutenu de près ou de loin et qui ne m'ont jamais oublié. Vous étiez différents mais chacun d'entre vous a su comment apporter le bonheur à ma vie.

Sans oublier mon binôme et ma meilleure amie "yousra" pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Aussi ma meilleure amie "Riham" qui nous a encouragés tout au long du projet.

"Amina"

Remerciement

En premier lieu, on tient à remercier notre Dieu ALLAH qui nous a donné la force pour achever ce projet.

Nous tenons à exprimer toute nos reconnaissances à notre encadrante Madame LABRAOUI Nabila. Nous la remercions de nous avoir encadrés, orienté, aidé et conseillé. Nous espérons être à la hauteur de sa confiance. Qu'elle trouve dans ce travail le fruit de ses efforts et l'expression de notre profonde gratitude.

Nous remercions en particulier Madame BENSAID Rajaa pour l'aide qu'elle nous a fournie et les connaissances qu'elle a su nous transmettre. Nous la remercions également pour sa disponibilité et la qualité de ses conseils.

Nous adressons nos remerciements les plus sincères aux membres du jury Monsieur MANA Mohamed et Monsieur BAMBRIK Ilyas qui ont bien voulu examiner ce modeste travail, et pour leur présence, leur lecture attentive de notre mémoire ainsi que pour les remarques qu'ils nous adresseront lors de la soutenance afin d'améliorer notre travail.

Nos remerciements s'adressent également à tous nos professeurs pour leur générosité et la patience dont ils ont su faire preuve malgré leurs charges professionnelles.

Nous tenons à adresser nos remerciements les plus chaleureux à l'égard de nos parents, nos frères et sœurs, nos amis, qui nous ont apporté leurs soutiens moral et intellectuel tout au long de notre cursus.

A tous ceux qui de loin ou de près, ont contribué par leurs conseils, leurs encouragements et leurs amitiés, à l'édification de ce modeste travail, trouvent ici l'expression de notre profonde reconnaissance.

RÉSUMÉ :

Le Fog Computing est un paradigme émergent qui étend les capacités du Cloud Computing en déployant des ressources informatiques et de stockage à la périphérie du réseau, plus près des utilisateurs et des objets connectés. Cependant Les environnements Fog sont souvent distribués, hétérogènes et dynamiques, ce qui rend la détection d'intrusion plus complexe par rapport aux systèmes traditionnels. Dans ce mémoire, nous proposons une approche basée sur l'apprentissage fédéré pour détecter les intrusions dans le Fog Computing. L'apprentissage fédéré est une technique d'apprentissage machine qui permet d'entraîner des modèles de manière collaborative sans partager les données brutes entre les nœuds. Cette approche préserve la confidentialité des données tout en permettant l'apprentissage d'un modèle global basé sur les informations locales de chaque nœud Fog. Les résultats obtenus démontrent le potentiel de l'apprentissage fédéré comme une solution prometteuse pour renforcer la sécurité des environnements Fog, ouvrant ainsi la voie à de futures recherches dans ce domaine.

Mots clés: Fog Computing, Cloud Computing, la détection d'intrusion, l'apprentissage fédéré, l'apprentissage machine

Abstract:

Fog Computing is an emerging paradigm that extends the capabilities of Cloud Computing by deploying computing and storage resources at the network edge, closer to users and connected devices. However, Fog environments are often distributed, heterogeneous, and dynamic, which makes intrusion detection more complex compared to traditional systems. In this thesis, we propose a federated learning-based approach for detecting intrusions in Fog Computing. Federated learning is a machine learning technique that allows models to be trained collaboratively without sharing raw data among nodes. This approach preserves data privacy while enabling the learning of a global model based on the local information of each Fog node. The obtained results demonstrate the potential of federated learning as a promising solution to enhance the security of Fog environments, thereby paving the way for future research in this field.

Keywords: Fog Computing, Cloud Computing, intrusion detection, Federated learning, machine learning

ملخص: حوسبة الضباب هي نموذج ناشئ يوسع قدرات الحوسبة السحابية من خلال نشر موارد الحوسبة والتخزين على حافة الشبكة ،بالقرب من المستخدمين والأجسام المتصلة. ومع ذلك، غالبًا ما تكون بيئات الضباب موزعة وغير متجانسة وديناميكية، مما يجعل اكتشاف التطفل أكثر تعقيداً من الأنظمة التقليدية. في هذا الموجز، نقترح نهج تعلم اتحادي لاكتشاف الاقتحامات في حوسبة الضباب. التعلم الفيدرالي هو أسلوب تعلم آلي يسمح لك بتدريب النماذج بطريقة تعاونية دون مشاركة البيانات الأولية بين العقد. يحافظ هذا النهج على سرية البيانات مع السماح بتعلم نموذج عالمي بناءً على المعلومات المحلية لكل عقدة ضباب. تظُهر النتائج إمكانات التعلم الفيدرالي كحل واعد لتعزيز سلامة بيئات الضباب، مما يمهد الطريق للبحث المستقبلي في هذا المجال.

الكلمات المفتاحية: حوسبة الضباب , الحوسبة السحابية , اكتشاف التطفل , التعلم الفيدرالي , علم آلي

Liste des Figures

FIGURE 1. 1 : LES MODELES DE SERVICES DE CLOUD COMPUTING.[2] 11
FIGURE 1. 2: THE HIERARCHICAL ARCHITECTURE OF FOG COMPUTING [10]. 16
FIGURE 1. 3: ARCHITECTURE DE SYSTEME PROPOSEE [12] 17
FIGURE 1. 4: IDS BASE SUR LES SIGNATURES [29] 23
FIGURE 1. 5: IDS BASE SUR LES ANOMALIES 24
FIGURE 1. 6 : IDS RESEAU [15] 25
FIGURE 1. 7 : IDS HYBRIDE [15] 25
FIGURE 2. 1: INTELLIGENCE ARTIFICIELLE, MACHINE LEARNING ET DEEP LEARNING [36] 29
FIGURE 2. 2 : L'ARCHITECTURE DE RESEAUX DE NEURONES [47] 33
FIGURE 2. 3: LA STRUCTURE DE RNN [51]. 34
FIGURE 2. 4: SCHEMA SIMPLIFIE D'UNE CELLULE D'UN RESEAU LSTM [53]. 36
FIGURE 2. 5: CELLULE D'UN RESEAU LSTM [53]. 37
FIGURE 2. 6: FONCTION SIGMOÏDE [56] 38
FIGURE 2. 7: FONCTION TANH [56]. 39
FIGURE 2. 8: FONCTION SOFTMAX [56]. 39
FIGURE 2. 9: FEDERATED LEARNING SYSTEM (HORIZONTAL) [67] 42
FIGURE 2. 10: ARCHITECTURE FOR A VERTICAL FEDERATED LEARNING SYSTEM. [68] 43
FIGURE 2. 11: FEDERATED TRANSFER LEARNING (FTL) ARCHITECTURE. [70] 43
FIGURE 2. 12: FEDERATED LEARNING BASED IDS (FLIDS) [73] 44
FIGURE 3. 1: ARCHITECTURE D'IDS BASÉ SUR FEDERATED LEARNING DANS LE FOG 48
FIGURE 3. 2 : PRE-PROCESSING ET NETTOYAGE DES DONNEES. 53
FIGURE 3. 3 : RANDOM FOREST CLASSIFIER. 54
FIGURE 3. 4 : LES MATRICES DE CONFUSION DE DEEP LEARNING ET FEDERATED LEARNING. 58
FIGURE 3. 5 : LES COURBE ROC 60
FIGURE 3. 6 : EVALUATION DES PERFORMANCES 62
FIGURE 3. 7 : LES MATRICES DE CONFUSION DE DEEP ET FEDERATED DEEP LEARNING. 63
FIGURE 3. 8: COURBES ROC DE FDL. 64
FIGURE 3. 9: COURBES ROC DE DL 64
FIGURE 3. 10 : EVALUATION DES PERFORMANCES. 65

Liste des Tableaux

TABLEAU 3. 1 : LES DONNEES SELECTIONNEES [76] 52
TABLEAU 3. 2: CATEGORIE DES ATTAQUES. 52
TABLEAU 3. 3: PARAMETRES DU MODELE LSTM UTILISE. 56
TABLEAU 3. 4: PARAMETRE DE COMPILATION DU MODELE. 56
TABLEAU 3. 5 : COMPARAISON ENTRE LES IDS BASE SUR LE DLET LES IDS BASE SUR LE FL. 61
TABLEAU 3. 6 : COMPARAISON ENTRE LES IDS BASE SUR LE DL ET LES IDS BASE SUR LE FL. 65

Introduction générale

Ces dernières années, le nombre d'appareils connectés à l'Internet des objets (IoT) a augmenté de façon spectaculaire dans différents domaines tels que les systèmes industriels, les villes intelligentes, les soins de santé et les transports. Avec l'adoption croissante des paradigmes du calcul en périphérie (edge computing) et du fog computing, le traitement et l'analyse de ces immenses quantités de données IoT sont devenus plus efficaces et réalisables au plus près de la source des données.

La sécurité des systèmes IoT est un enjeu majeur de cet environnement informatique périphérique distribué. Les systèmes de détection d'intrusions (IDS) jouent un rôle important pour la sécurité du réseau, qui est la première ligne de défense dans le réseau. Les solutions IDS classiques sont généralement basées sur des architectures centralisées, dans lesquelles toutes les données sont envoyées à un serveur central pour analyse et détection. Cependant, dans le fog, le traitement centralisé des données peut être inefficace en raison du volume élevé de données IoT et de la bande passante limitée.

Pour relever ce défi, le concept d'apprentissage fédéré est apparu comme une approche prometteuse pour la construction de systèmes IDS robustes et efficaces dans les environnements du fog computing. L'apprentissage fédéré permet de former des modèles de machine learning en collaboration sur plusieurs périphériques, sans avoir besoin de transmettre des données sensibles à un serveur central. Au lieu de cela, le processus de formation du modèle se déroule localement sur les appareils en périphérie, et seules les mises à jour agrégées du modèle sont partagées avec une autorité centrale.

En utilisant l'apprentissage fédéré, les IDS dans le fog computing peuvent bénéficier de plusieurs avantages, il renforce la confidentialité et la sécurité des données, et réduit les surcharges de communication et les exigences de bande passante en effectuant la formation du modèle localement et en partageant uniquement les mises à jour du modèle. Cela améliore considérablement l'efficacité et la scalabilité du système IDS dans les environnements du fog computing. De plus, l'apprentissage fédéré en IDS permet une meilleure adaptabilité aux environnements IoT dynamiques.

En conclusion, l'intégration des systèmes de détection d'intrusion (IDS) avec un apprentissage fédéré dans les environnements de calcul de brouillard offre une solution prometteuse pour relever les défis de sécurisation des systèmes IoT. En tirant parti de la puissance de l'apprentissage automatique distribué, l'apprentissage fédéré permet de préserver la vie privée et de déployer des IDS efficaces à

la périphérie du réseau. Cette approche assure une détection efficace des menaces, minimise la transmission de données et améliore l'évolutivité et l'adaptabilité des systèmes IDS dans les architectures de calcul de brouillard.

Dans ce travail de fin d'étude nous avons proposé un système de détection d'intrusion pour le fog computing, basé sur l'apprentissage fédéré.

Structure du mémoire :

Notre mémoire est constitué de trois chapitres comme suivant:

1) **Chapitre 1 : Les systèmes de détection d'intrusion dans le fog computing**

Dans le premier chapitre, nous avons défini le cloud computing et comment le fog computing est venu pour résoudre les problèmes du cloud. Ensuite nous avons mentionné les problèmes de sécurité du fog computing et effectué une description détaillée sur les attaques dans le fog computing. Enfin nous avons présenté quelques solutions de sécurité en mettant l'accent sur les IDS et expliquant tous ses concepts.

2) **Chapitre 2 : Federated Learning**

Dans le second chapitre, nous avons dressé une vue globale de l'intelligence artificielle. Par la suite, nous avons présenté en détail la machine learning, le deep learning et le federated learning.

3) **Chapitre 3 *:* Modélisation d'un modèle d'apprentissage profond**

Le troisième chapitre est consacré à la modélisation du modèle d'apprentissage profond, tout d'abord nous avons expliqué le pré-processing et comment en a nettoyé notre data, puis nous avons utilisé le deep learning et le federated learning pour la détection d'intrusion et ensuite fait une comparaison entre les deux résultats.

Table des matières

LISTE DES FIGURES **1**

LISTE DES TABLEAUX **2**

INTRODUCTION GENERALE **3**

CHAPITRE 1 : LES SYSTEMES DE DETECTION D'INTRUSION DANS LE FOG COMPUTING **9**

INTRODUCTION **9**

1. CLOUD COMPUTING **10**

1.1. Définition *10*

1.2. Les différents services du Cloud Computing *10*

1.2.1. Software as a Service (SaaS) 11

1.2.2. Platform as a Service (PaaS) 11

1.2.3. Infrastructure as a Service (IaaS) 11

1.3. Les caractéristiques *12*

1.4. Problèmes de Cloud Computing *12*

2. FOG COMPUTING **13**

2.1. Définition *13*

2.2. Caractéristiques essentielles du Fog Computing *14*

2.3. Architecture *14*

2.3.1. La Couche IoT /Utilisateur (Couche terminale) 15

2.3.2. La Couche Fog 15

2.3.3. La Couche Cloud 16

2.4. Le Fog computing et les applications IoT *16*

2.4.1. Smart Grid 16

2.4.2. Pression de l'eau aux barrages 17

2.4.3. Smart Utility Service 17

2.4.4. Données sur la santé 17

2.4.5. Maison intelligente 17

3. LE PROBLEME DE SECURITE DU FOG COMPUTING **18**

4. LES ATTAQUES DANS LE FOG COMPUTING **19**

4.1. Attaque de réseau *19*

4.2. Attaque de services *19*

4.2.1 Déni de service (DoS) 19

4.2.2 Déni de service distribué (DDoS) 19

4.2.3 Attaques d'injection de code 19

4.2.4 Attaques d'usurpation d'identité 20

4.2.5 Attaques de falsification de données 20
5. TECHNIQUES DE SECURITE DANS LE FOG COMPUTING 20
5.1. L'Authentification 20
5.2. Le Chiffrement 20
5.3. Le Contrôle d'accès 20
5.4. Le Pare-feu 21
5.5. Détection d'intrusion (IDS) 21
5.6. La Mise à jour du micrologiciel 21
5.7. Sécurité physique 21
6. SYSTEMES DE DETECTION D'INTRUSION (IDS) 21
6.1. Définition 21
6.2. Les Types d'IDS 22
6.2.1. IDS basés sur les signatures 22
6.2.2. IDS basés sur les anomalies 23
6.2.3. IDS Réseau 23
6.2.4. IDS basés sur l'hôte 24
6.2.5. IDS hybrides 24
CONCLUSION 25
CHAPITRE 2 : FEDERATED LEARNING 26
INTRODUCTION 26
1. L'INTELLIGENCE ARTIFICIELLE 26
1.1. Définition 26
1.2. L'impact de l'Intelligence Artificielle 27
1.3. La relation entre l'IA, ML et DL 27
2. MACHINE LEARNING 28
2.1. Définition 28
2.2. Types d'apprentissage automatique 28
2.2.1. Apprentissage supervisé 29
2.2.2. Apprentissage non supervisé 29
2.2.3. Apprentissage par renforcement 29
2.3. Processus de l'apprentissage automatique 29
Étape 1 : Sélectionner et préparer l'ensemble de données d'entraînement 29
2.4. IDS basés sur le Machine Learning 30
3. LE DEEP LEARNING 31
3.1. Définition 31
3.2. Définition de réseaux de neurones 31
3.3. Fonctionnement de réseaux de neurones 31

3.4. Types des réseaux de neurones *32*
3.4.1. Les réseaux de neurones dit "feed-forward'' (à propagation avant) 33
3.4.2. Les réseaux de neurones récurrents "feed-back'' 33
3.5. LSTM (Long Short-Term Memory) *34*
3.5.1. Architecture des LSTM 34
3.5.2. Les Étapes LSTM 35
3.6. Fonction d'activation *36*
3.6.1. Sigmoid 37
3.6.2. tanh 37
3.6.3. Softmax 38
3.7. Fonction de perte (loss function) 38
3.8. Optimiseur 39
3.9. Cross validation 39
3.10. Early stopping 40
3.11. La détection d'intrusion basée sur le Deep learning 40
4. FEDERATED LEARNING 40
4.1. Définition *41*
4.2. Fonctionnalité *41*
4.3. Types *41*
4.3.1. Apprentissage fédéré horizontalement (HFL) 41
4.3.2. Apprentissage fédéré verticalement (VFL) 42
4.3.3. Apprentissage fédéré par transfert (FTL) 42
4.4. Application *43*
4.6. L'approche Centralisée *43*
CONCLUSION 44
CHAPITRE 3 : MODELISATION D'UN MODELE D'APPRENTISSAGE PROFOND. 45
Introduction *45*
1. PROBLEMATIQUE 45
2. SOLUTION PROPOSEE 46
3. OBJECTIFS DE PROJET 46
4. L'ARCHITECTURE DE PROJET 46
5. ENVIRONNEMENT DE TRAVAIL 48
5.1. Environnement matériel *48*
5.2. Environnement d'exécution *48*
5.2.1. Google Colab 48
5.2.2. Langage utilisé 49
5.2.3. Bibliothèques et Frameworks utilisées 49
6. PRESENTATION DU DATASET 49

7. METHODOLOGIE **52**
7.1. Modélisation *52*
7.2. Modèle LSTM (Long Short-Term Memory) *54*
8. SCENARIOS **56**
8.1. Scénario 1 : Deep learning *56*
8.2. Scénario 2 : Federated deep learning *56*
9. METRIQUE D'EVALUATION (BINAIRE) **57**
9.1. Matrice de confusion *57*
9.2. La courbe ROC (Receiver Operating Characteristic) *58*
9.3. Performance *59*
9.3.1 Précision 59
9.3.2. Rappel (recall) 59
9.3.3. Score F1 (F1-score) 59
9.3.4. exactitude (Accuracy) 59
9.3.5 Evaluation des résultats de performance 59
10. METRIQUE D'EVALUATION (MULTI-CLASSE) **61**
10.1. Matrice de confusion *61*
10.2. Courbe ROC (Receiver Operating Characteristic) *61*
10.3. Evaluation des résultats *62*
CONCLUSION **65**
CONCLUSION GENERALE **66**
REFERENCES **67**

CHAPITRE 1 :

Les Systèmes de Détection d'Intrusion dans le Fog Computing

Introduction

Le fog computing fait partie du paradigme de l'informatique en nuage (cloud) qui rapproche le nuage de la périphérie du réseau. Bien que le fog et le cloud utilisent des ressources similaires (traitement, stockage et réseau) et utilisent les mêmes mécanismes et attributs (virtualisation, multi-entité), le fog computing apporte de nombreux avantages aux appareils IoT (terminaux finaux).

Ce paradigme est ajouté pour combler les lacunes du cloud. Le Fog computing est défini comme "un scénario dans lequel un grand nombre d'appareils sans fil hétérogènes sont connectés ensemble dans un réseau, communiquent et coopèrent potentiellement entre eux et avec le réseau pour effectuer des tâches de stockage et de traitement sans l'intrusion de tiers".

La sécurité dans le Fog Computing est essentielle pour assurer la confidentialité, l'intégrité et la disponibilité des données. Cela nécessite des mesures de sécurité robustes pour protéger les données et les systèmes contre les attaques malveillantes, les fuites de données et les pertes de données. Des stratégies de sécurité efficaces doivent être mises en place pour garantir que les utilisateurs peuvent utiliser le Fog Computing en toute confiance, sans compromettre leur sécurité ou leur vie privée.

Dans ce chapitre, nous présenterons en premier lieu une description globale de la technologie du cloud computing, ses différents services, ses avantages et ses limites. Ensuite nous présenterons le paradigme du fog computing comme une solution aux inconvénients du cloud. Nous aborderons également la problématique de sécurité dans ce nouveau paradigme et nous nous concentrerons sur les systèmes de détection d'intrusion comme mesure de sécurité.

1. Cloud computing

1.1. Définition

Le Cloud signifie « nuage » et Computing « informatique ». Le cloud computing ou informatique en nuage est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée. Plusieurs définitions du Cloud Computing existent mais nous nous souviendrons de celle du NIST (National Institute of Standards and Technology), qui définit le Cloud Computing comme étant un modèle permettant l'accès omniprésent, pratique et à la demande à un réseau partagé de ressources informatiques configurables (p. ex., réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement fournis et diffusés avec un minimum d'effort de gestion ou d'interaction avec les fournisseurs de services.[1]

1.2. Les différents services du Cloud Computing

Le Cloud computing est une technologie dans laquelle les ressources matérielles et logicielles telles que les applications spéciales, le processeur, le stockage et bien d'autres sont fournis aux utilisateurs en tant que services payants. Il existe plusieurs modèles de service pour le Cloud computing, selon la façon dont ce service est fourni à l'utilisateur, le degré de contrôle que l'utilisateur a sur les ressources et le type de ressources que l'utilisateur a demandées [2]. L'institut national des normes et de la technologie (NIST) définit trois modèles de services de base, à savoir IaaS, PaaS et SaaS comme la montre la Figure 1.1 suivante :

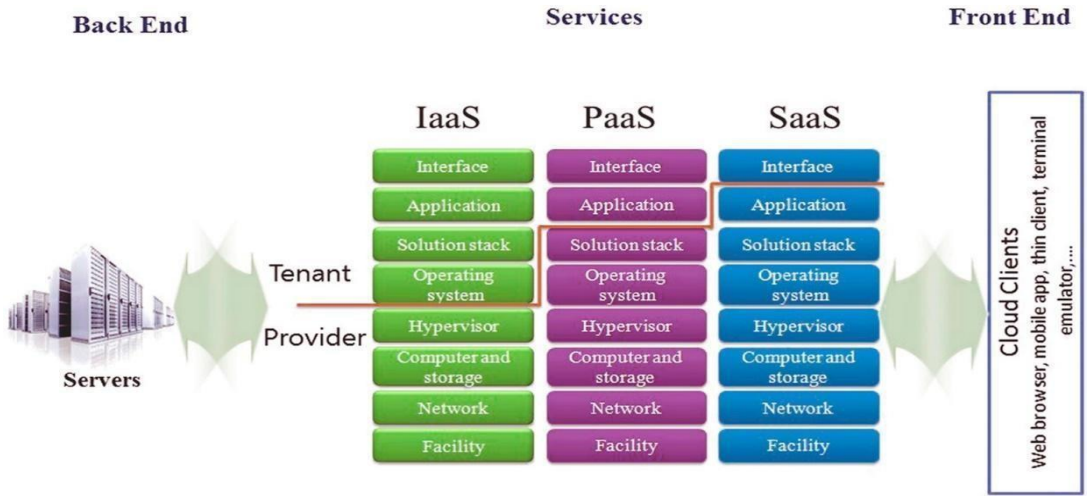

Figure 1. 1 : Les modèles de services de Cloud computing.[2]

1.2.1. Software as a Service (SaaS)

L'acronyme « SAAS » est le plus connu dans le monde du Cloud Computing. Sa signification est l'application en tant que service. La capacité fournie à l'utilisateur d'utiliser les applications du fournisseur fonctionnant sur une infrastructure cloud. Les utilisateurs peuvent accéder aux applications à partir de divers appareils clients au moyen d'une interface client allégée, comme un navigateur Web (p. ex., courriel Web), ou d'une interface de programme. L'utilisateur ne gère ni ne contrôle l'infrastructure cloud sousjacente, y compris le réseau, les serveurs, les systèmes d'exploitation, le stockage ou même les capacités d'applications individuelles, sauf peut-être de paramètres de configuration d'applications spécifiques aux utilisateurs [2] [3].

1.2.2. Platform as a Service (PaaS)

Elle est également connue sous le nom de « CloudWare ». La capacité offerte au consommateur consiste à déployer sur l'infrastructure du cloud des applications créées ou acquises par les utilisateurs à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur. L'utilisateur ne gère ni ne contrôle l'infrastructure du cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation ou le stockage, mais il offre des facilités à gérer le déroulement des opérations lors de la conception, du développement, du test, du déploiement et de l'hébergement d'applications web à travers des outils et des services tels que: Le travail collaboratif (« team collaboration ») et l'intégration des services web et bases de données. Ces services sont fournis au travers une solution complète destinée aux développeurs et disponible immédiatement via l'internet [2] [3].

1.2.3. Infrastructure as a Service (IaaS)

L'IAAS (Infrastructure as a Service) est un modèle qui permet de fournir des infrastructures informatiques en tant que service. Ce terme était originellement connu sous le nom de (Hardware as a Service). L'IaaS est la fourniture aux utilisateurs des ressources de traitement, de stockage, de réseaux et d'autres ressources informatiques fondamentales où l'utilisateur est en mesure de déployer et d'exécuter des logiciels arbitraires, qui peuvent inclure des systèmes d'exploitation et des applications. L'utilisateur ne gère ni ne contrôle l'infrastructure du cloud sous-jacente, mais il a le contrôle des systèmes d'exploitation, du stockage et des applications déployées, et peut-être un contrôle limité de certains composants réseau (p.
ex., les pare-feu hôtes). L'un des principaux avantages de l'IaaS est le système de paiement basé sur l'utilisation. Cela permet aux clients de payer au fur et à mesure de leur croissance [2] [3].

1.3. Les caractéristiques

NIST, dans sa dernière définition du Cloud computing, a mis en évidence cinq caractéristiques essentielles qui toutes les technologies infonuagiques devraient être intégrées [1]. Ces caractéristiques sont détaillées comme suit :

- **Libre-service à la demande :** Les utilisateurs des services de Cloud attendent un accès à la demande et presque instantané aux ressources.

- **Un accès au réseau :** les services de Cloud et les ressources sont disponibles sur le réseau, et accessibles au moyen de mécanismes et des protocoles standard qui favorisent leur utilisation à partir des plates-formes client hétérogènes telles que les ordinateurs portables, les postes de travail, les téléphones mobiles et les tablettes.

- **La mutualisation des ressources informatiques :** Les ressources informatiques du fournisseur de cloud sont regroupées pour servir plusieurs utilisateurs en utilisant un modèle multi-locataires. Ces ressources physiques et virtuelles assignées dynamiquement et réaffectées en fonction de la demande des clients qui n'ont généralement aucun contrôle ou connaissance de l'emplacement exact des ressources fournies.

- **Service Mesuré :** Les systèmes du cloud contrôlent et optimisent automatiquement l'utilisation des ressources afin de mesurer leur consommation avec un niveau d'abstraction approprié. L'utilisation des ressources peut être surveillée, contrôlée et déclarée, ce qui assure la transparence tant pour le fournisseur que pour le consommateur du service utilisé.

- **Élasticité rapide :** Les ressources peuvent être fournies de façon élastique et libérées, dans certains cas automatiquement, pour s'étendre rapidement vers l'extérieur et vers l'intérieur en fonction de la demande. Cela donne aux consommateurs l'illusion de disposer de quantités infinies de sources pouvant être allouées à tout moment.

1.4. Problèmes de Cloud Computing

En dépit de tous les avantages comme mentionné ci-dessus le Cloud computing a également ses inconvénients. Certaines entreprises préfèrent ne pas l'utiliser pour les raisons techniques et légales suivantes:

- **Sécurité des données :** Dans le passé proche, il y a eu pas mal de fois de fuite massive de données dans le stockage ICloud, où les comptes ICloud de plusieurs célébrités ont été piratés et leurs photos privées affichées en ligne. C'est le plus grand désavantage du cloud – vous mettez vos données en ligne où d'autres

personnes peuvent y accéder en cas de violation. C'est la raison principale pour laquelle de nombreuses entreprises hésitent à migrer vers le cloud et malgré son succès [4].

- **Temps d'arrêt :** la plupart des services informatiques sur le cloud demeurent disponibles 24h/24h, mais pour certains services il est prévu parfois du temps mort pour lesquels les services ne sont plus accessibles. Cela peut être dû à la maintenance périodique ou comme expliqué précédemment, parfois le service le fournisseur ne s'engage disponible que pour une période limitée par jour [4].
- **Contrôle limité :** Les consommateurs ont très peu de contrôle sur leurs produits dans le cloud [4].
- **Dépendance à l'égard du réseau :** un autre inconvénient majeur du cloud est sa dépendance à l'Internet. Une connexion fiable est nécessaire au bon fonctionnement [4].
- **Latence :** Les nouvelles applications dans le scénario IoT ont des exigences élevées en temps réel. Dans le modèle de cloud computing, les applications envoient des données au centre de données et obtiennent une réponse, ce qui augmente la latence du système. Par exemple, les véhicules de conduite autonome à grande vitesse nécessitent des millisecondes de temps de réponse. De graves conséquences se produisent une fois que la latence du système dépasse les attentes, en raison de problèmes de réseau [5].

2. Fog computing

2.1. Définition

Fog computing est un terme créé par Cisco en 2014, désigne la décentralisation d'une infrastructure informatique en étendant le nuage grâce à la mise en place de nœuds stratégiques entre le nuage et les périphériques. Cela rapproche les données, le calcul, le stockage et les applications de l'utilisateur ou du dispositif IoT où les données doivent être traitées, créant ainsi un brouillard hors du nuage centralisé et réduisant les délais de transfert de données nécessaires au traitement des données [6].

L'informatique par brouillard (fog computing) est un modèle à plusieurs niveaux permettant un accès omniprésent à un continuum partagé de ressources informatiques évolutives. Le modèle facilite le déploiement d'une latence distribuée applications et services, et se compose de nœuds de brouillard (physiques ou virtuels), résidant entre les dispositifs d'extrémité intelligents (smart end-devices) et les services centralisés (cloud) [7].

2.2. Caractéristiques essentielles du Fog Computing

Les caractéristiques suivantes sont essentielles pour distinguer le fog computing des autres paradigmes informatiques. Toutefois, un utilisateur d'un dispositif terminal intelligent ou de l'IoT n'est pas tenu d'utiliser toutes les caractéristiques lors de la consommation d'un service de fog computing.

- **Connaissance contextuelle de la localisation et faible latence :** Le calcul du brouillard offre la latence la plus faible possible en raison de la connaissance par les nœuds de brouillard de leur emplacement logique dans le contexte de l'ensemble des systèmes et des coûts de latence pour communiquer avec d'autres nœuds. Comme les nœuds de brouillard sont souvent situés au même endroit que les dispositifs d'extrémité intelligents, l'analyse et la réponse aux données générées par ces dispositifs sont beaucoup plus rapides qu'à partir d'un service de nuage centralisé ou d'un centre de données [7] [8].

- **Répartition géographique :** Contrairement au Cloud plus centralisé, les services et les applications visés par le fog exigent des déploiements distribués largement, mais géographiquement identifiables. Le fog, par exemple jouera un rôle actif dans la prestation de services de diffusion en continu de haute qualité aux véhicules en mouvement, grâce à des proxies et des points d'accès géographiquement situés le long des autoroutes et des voies [7] [8].
- **Hétérogénéité :** le brouillard informatique prend en charge la collecte et le traitement des données de différents facteurs de forme acquis par de multiples types de capacités de communication réseau [7] [8].
- **Interopérabilité et fédération :** La prise en charge transparente de certains services (les services de streaming en temps réel en sont un bon exemple) nécessite la coopération de différents fournisseurs. Par conséquent, les composants de calcul de brouillard doivent être capables d'inter-fonctionner, et les services doivent être fédérés entre les domaines [7] [8].
- **Interactions en temps réel :** Les applications de calcul de brouillard impliquent des interactions en temps réel plutôt que le traitement par lots [7] [8].
- **Soutien à la mobilité :** Il est essentiel pour de nombreuses applications Fog de communiquer directement avec les appareils mobiles, et donc de soutenir les méthodes de mobilité, comme le protocole LISP, qui découplent l'identité de l'hôte de l'identité de localisation, et nécessitent un système d'annuaire distribué [7] [8].

2.3. Architecture

Le modèle de référence de l'architecture informatique du brouillard est principalement dérivé de la structure fondamentale à trois couches [9]. La figure 1.2 montre l'architecture hiérarchique du calcul du brouillard.

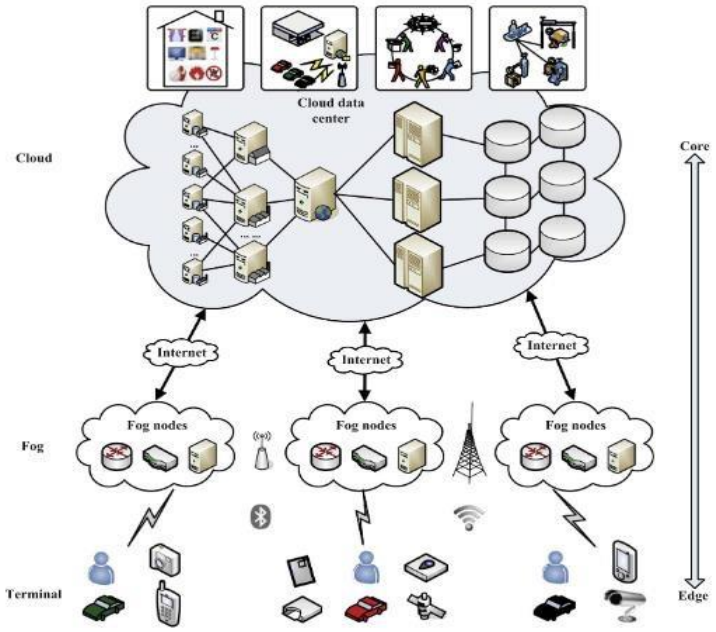

Figure 1. 2: The hierarchical architecture of fog computing [10].

L'architecture hiérarchique est composée des trois couches suivantes:

2.3.1. La Couche IoT /Utilisateur (Couche terminale)

La couche la plus proche de l'utilisateur final et de l'environnement physique. Il se compose de divers appareils IoT, par exemple, les capteurs, téléphones mobiles, véhicules intelligents, cartes à puce, lecteurs, et ainsi de suite. Ces dispositifs sont largement répartis géographiquement en général. Ils sont chargés de détecter les caractéristiques des objets physiques ou des événements et de transmettre ces données détectées à la couche supérieure pour traitement et stockage [9].

2.3.2. La Couche Fog

Cette couche est située sur le bord du réseau, elle est composée d'un grand nombre de nœuds de brouillard, qui comprennent généralement des routeurs, des passerelles, des commutateurs, des points d'accès, des stations de base, des serveurs de brouillard spécifiques, etc. Ces nœuds de brouillard sont largement répartis entre les dispositifs d'extrémité et le nuage, Par exemple, les cafés, les centres commerciaux, les terminaux d'autobus, les rues, les parcs, etc. Ils peuvent être statiques à un emplacement fixe, ou mobile sur un transporteur en mouvement. Les dispositifs d'extrémité peuvent facilement se connecter avec des nœuds de brouillard pour obtenir des services. Ils ont les capacités de calculer, transmettre et stocker temporairement les données captées [9].

2.3.3. La Couche Cloud

La couche cloud se compose de plusieurs serveurs et dispositifs de stockage haute performance, et fournit divers services d'application, tels que la maison intelligente, le transport intelligent, usine intelligente, etc. Elle dispose de puissantes capacités de calcul et de stockage pour prendre en charge une analyse de calcul étendue et le stockage permanent d'une énorme quantité de données [9].

2.4. Le Fog computing et les applications IoT

La plateforme de calcul du brouillard a un large éventail d'applications. Dans cette section, nous aborderons plusieurs applications fascinantes qui bénéficieront de l'informatique par brouillard.

La figure 1.3 présente l'architecture informatique et d'information idéalisée soutenant les futures applications IoT et illustre le rôle de l'informatique par brouillard.

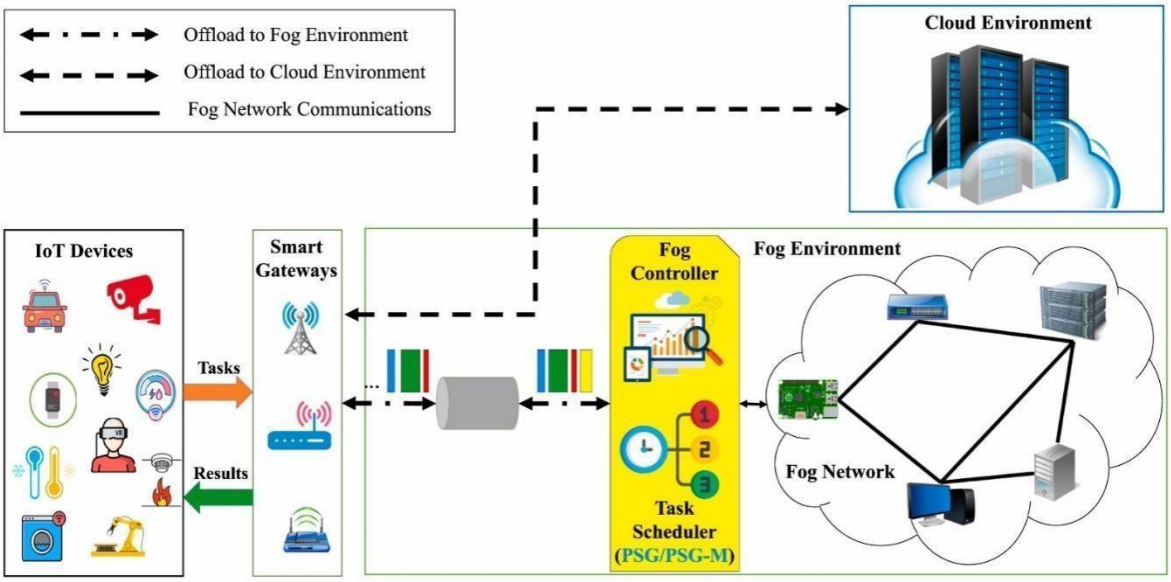

Figure 1. 3 : Architecture de système proposée [12]

2.4.1. Smart Grid

De nombreuses applications de l'Internet des objets (IoT) seront déployées dans le réseau intelligent pour permettre un fonctionnement efficace du réseau. Les applications IoT dans le réseau intelligent pourraient être utilisées pour surveiller la ligne de transport d'électricité, la sous-station, gérer la recharge/décharge des véhicules électriques, recueillir des informations auprès des utilisateurs, etc. [13]

2.4.2. Pression de l'eau aux barrages

Les capteurs installés dans les barrages envoient les données au cloud où elles sont analysées et les responsables sont alertés en cas de divergence. Le problème qui se pose ici est le retard de l'information qui pourrait être dangereuse. Pour résoudre ce problème, le Fog est utilisé, et puisqu'il est proche des systèmes d'extrémité, il est plus facile de transmettre des données, de les analyser et de donner une rétroaction instantanée [11].

2.4.3. Smart Utility Service

L'objectif principal est de conserver le temps, l'argent et l'énergie. L'analyse des données doit fonctionner chaque minute pour rester à jour. Cela implique principalement les utilisateurs finaux, donc le cloud pourrait ne pas servir l'objectif. Ces applications informent les utilisateurs tous les jours sur les appareils qui économisent plus d'énergie. IOT crée également beaucoup de trafic dans le réseau où l'envoi d'autres données est difficile donc le fog est une bonne alternative [11].

2.4.4. Données sur la santé

La gestion des données sur la santé est une question délicate puisque les données sur la santé contiennent des renseignements précieux et privés. Avec le brouillard informatique, il est en mesure de réaliser l'objectif que le patient prendra possession de ses propres données de santé localement. Ces données de santé seront stockées dans un nœud de brouillard comme un téléphone intelligent ou un véhicule intelligent. Le calcul sera externalisé de manière privée lorsque le patient sollicite l'aide d'un laboratoire médical ou d'un cabinet médical. La modification des données se produit directement dans le nœud de brume appartenant au patient [14].

2.4.5. Maison intelligente

Avec le développement rapide de l'Internet des objets, de plus en plus d'appareils intelligents et de capteurs sont connectés à la maison. Cependant, les produits de différents fournisseurs sont difficiles à travailler ensemble. Certaines tâches, qui nécessitent une grande quantité de calcul et de stockage, par exemple l'analyse vidéo en temps réel, sont infaisables en raison de la capacité limitée du matériel. Pour résoudre ces problèmes, le brouillard informatique est utilisé pour intégrer tous les débris dans une seule plate-forme et habiliter ces applications Smart Home avec des ressources élastiques [14].

3. Le problème de sécurité du fog computing

Le Fog offre plusieurs avantages, notamment la simplicité de déploiement, l'accessibilité, l'évolutivité, la fiabilité, la tolérance aux pannes, les ressources partagées, le stockage étendu la capacité et les économies de coûts. Si le Fog computing offre de nombreux avantages, il introduit une multitude de problèmes de sécurité et de failles pour les fournisseurs de services fog et utilisateurs. Les fournisseurs de services de fog computing doivent résoudre les problèmes de sécurité typiques associés aux réseaux de communication conventionnels. En même temps, ils devraient résoudre d'autres problèmes inhérents au paradigme du cloud. le Fog Computing a rencontré plusieurs obstacles .Voici quelques-unes des difficultés du Fog Computing.

- **La Confiance**

Aborder les problèmes liés à la confiance dans un réseau de fog est légèrement plus délicat par rapport à l'environnement de cloud computing. L'ouverture de l'environnement informatique de fog et le bidirectionnel l'exigence de confiance sont des défis majeurs dans la conception d'un modèle pour réseau de brouillard [15].

- **La Disponibilité**

Cela garantit que les données doivent être accessibles à tout moment et que les parties autorisées doivent y avoir accès si nécessaire [16].

- **La Confidentialité**

Cela garantit que les données ne sont divulguées qu'aux entités ayant droit [17].

- **L'Intégrité**

Le processus de maintien de la cohérence et de l'exactitude des données est appelé intégrité. Les fournisseurs de cloud ou du fog doivent prendre des précautions pour éviter Modification non autorisée des données stockées [17].

- **L'Authentication**

Le mot " Authentication " signifie empêcher l'accès non autorisé à des informations protégées [15].

- **Le contrôle d’accès**

C'est un outil important pour la sécurité du système et la protection de la vie privée des utilisateurs. Le contrôle d'accès traditionnel répond généralement sous le même domaine de confiance. Suite à la nature externalisée du cloud computing et du fog, les contrôles d'accès au cloud computing ont été incorporés de manière cryptographique dans les informations externalisées [18].

4. Les Attaques dans le Fog Computing

Les attaques dans le fog computing peuvent être similaires à celles du cloud computing, mais nécessitent une attention supplémentaire en raison de la répartition des ressources informatiques et de l'implication des appareils des utilisateurs.

4.1. Attaque de réseau

Le fog utilise divers réseaux de communication pour interconnecter leurs éléments : du sans-fil réseaux vers les réseaux centraux mobiles et Internet. Un attaquant peut cibler ces vulnérabilités du cadre de communication [19].

4.2. Attaque de services

Les attaques de services dans le fog computing font référence à des tentatives malveillantes visant à perturber, compromettre ou exploiter les services et les infrastructures du fog computing. Parmi ces attaques [20], nous citons :

4.2.1 Déni de service (DoS)

Les attaquants cherchent à submerger les ressources du fog computing en générant un trafic excessif ou en exploitant des vulnérabilités pour empêcher les utilisateurs légitimes d'accéder aux services [20].

4.2.2 Déni de service distribué (DDoS)

Les attaquants utilisent un réseau de machines infectées (botnet) pour lancer une attaque coordonnée et déstabiliser les services du fog computing [20].

4.2.3 Attaques d'injection de code

Les attaquants tentent d'introduire du code malveillant dans les services du fog computing, ce qui peut entraîner des failles de sécurité, des interruptions de service ou des compromissions de données [20].

4.2.4 Attaques d'usurpation d'identité

Les attaquants cherchent à se faire passer pour des utilisateurs légitimes, des appareils ou des services du fog computing afin d'accéder à des ressources sensibles ou de compromettre l'intégrité des données [20].

4.2.5 Attaques de falsification de données

Les attaquants modifient ou altèrent les données transitant par les services du fog computing, ce qui peut entraîner des erreurs, des manipulations ou des décisions erronées basées sur des informations incorrectes [20].

5. Techniques de sécurité dans le Fog computing

Maintenir un niveau élevé de sécurité dans le Fog Computing est un défi constant qui exige une surveillance régulière et des mises à jour fréquentes afin de garantir la protection des données et des périphériques. Voici quelques techniques utilisées dans le Fog Computing :

5.1. L'Authentification

La première étape pour sécuriser le Fog Computing consiste à s'assurer que les utilisateurs et les périphériques sont authentiques. Des protocoles d'authentification robustes, tels qu'OAuth et OpenID Connecte, peuvent être utilisés pour s'assurer que seules les personnes et les périphériques autorisés ont accès aux ressources [21].

5.2. Le Chiffrement

Le paradigme de chiffrement a pour vocation d'assurer la confidentialité et l'authenticité des données. De nouvelles solutions cryptographiques apportent des réponses viables aux besoins du fog comme l'utilisation du chiffrement par attributs pour répondre au besoin du contrôle d'accès aux données et le chiffrement homomorphique pour le calcul sécurisé sur celles-ci. [22]

5.3. Le Contrôle d'accès

Le contrôle d'accès est une forme de stratégie de sécurité qui vise à limiter qui a accès **à** utilise**r** des ressources dans un environnement informatique. Le contrôle d'accès physique et logique est les deux formes de contrôle d'accès. Les systèmes de contrôle d'accès physique régissent les entités physiques et des installations telles que des campus, des bureaux, des salles et des ressources informatiques physiques.

Plusieurs techniques sont utilisées. On cite les blockchain comme technique très puissante dans le contrôle d'accès [23].

5.4. Le Pare-feu

Un pare-feu s'utilise à la frontière d'une zone. Les règles peuvent par exemple limiter le trafic à un seul serveur, ou l'autoriser uniquement avec certains types de poste, comme les postes de maintenance. La sécurité apportée par un pare-feu est étroitement liée à la précision des règles de configuration [24].

5.5. Détection d'intrusion (IDS)

Actuellement, les méthodes de détection d'intrusion sont largement employées pour minimiser les attaques et peuvent être appliquées à divers systèmes. Dans le Fog Computing, les IDS doivent être déployés à tous les niveaux de l'architecture à trois niveaux, pour surveiller et analyser le trafic et le comportement des nœuds Fog, des terminaux et des serveurs Cloud [15].

5.6. La Mise à jour du micrologiciel

Les mises à jour pourraient être automatisées. Comme c'est le cas pour plusieurs systèmes d'exploitation. La mise au micrologiciel peut être utilisée pour corriger les vulnérabilités de sécurité dans les périphériques [25].

5.7. Sécurité physique

La sécurité physique est importante pour protéger les ressources matérielles contre les accès non autorisés. Les mesures de sécurité physique telles que la vidéosurveillance et les serrures à clé peuvent être utilisées pour protéger les centres de données et les périphériques [26].

6. Systèmes de détection d'intrusion (IDS)

Nous verrons dans cette section les systèmes de détection d'intrusion (IDS) ainsi que leur type.

6.1. Définition

L'intrusion peut être définie comme tout type d'activités non autorisées qui causent des dommages à un système d'information. L'intrus ou l'attaquant essayant de trouver un moyen d'obtenir un accès non autorisé aux informations, de causer des dommages ou de se livrer à d'autres activités malveillantes. [27]

Un IDS ou un système de détection d'intrusion est un logiciel ou système matériel qui identifie les actions malveillantes sur les systèmes informatiques afin de permettre la sécurité du système maintenu, ce terme a été utilisé pour la première fois par James Anderson dans la fin des années 70 et le début des années 80 [28]. L'objectif d'un IDS est d'identifier différents types de trafic réseau malveillant et l'utilisation de l'ordinateur, qui ne peut pas être identifiée par un pare-feu traditionnel.

L'IDS à l'intérieur d'un nœud de fog réduit la latence entre les périphériques et les serveurs cloud. Ainsi, les données de trafic réseau normales provenant d'appareils haut de gamme sont analysées et surveillées, et les attaques malveillantes sont détectées dans la couche de fog [28].

6.2. Les Types d'IDS

Il existe plusieurs types d'IDS, classés selon leur fonctionnement et leur emplacement dans le réseau, on peut distinguer deux grandes catégories d'IDS :

6.2.1. IDS basés sur les signatures

La détection des abus est également appelée détection basée sur la signature. L'idée de base pour représenter les comportements d'attaque sous forme de signatures. Le processus de détection correspond aux signatures des échantillons à l'aide d'une base de données de signatures. Le principal problème dans la construction de systèmes de détection d'abus est de concevoir des signatures efficaces. Elle a un faible taux de fausses alarmes et qu'elle signale en détail les types d'attaques ainsi que les raisons possibles [29].

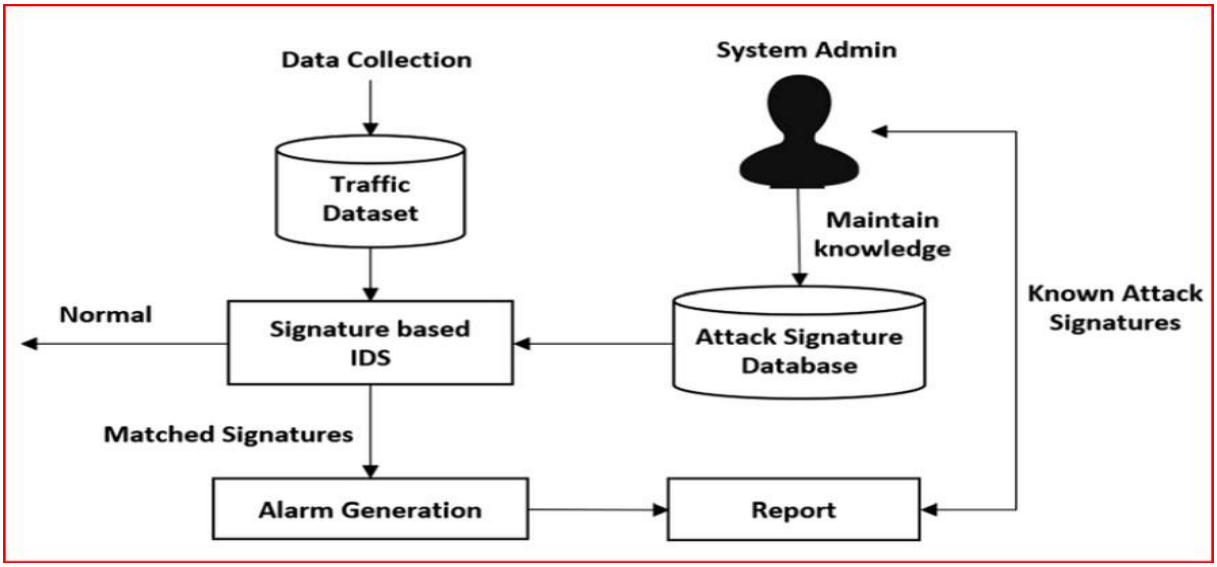

Figure 1. 4: IDS basé sur les signatures [29]

6.2.2. IDS basés sur les anomalies

Contrairement aux IDS basés sur les signatures, qui cherchent à identifier les attaques connues en comparant le trafic réseau à une base de données de signatures d'attaques connues, les IDS basés sur les anomalies utilisent une méthode statistique pour identifier les activités qui ne correspondent pas au comportement habituel du système ou du réseau. Ces systèmes surveillent le trafic réseau en temps réel et analysent les données pour détecter les modèles de trafic inhabituels [30].

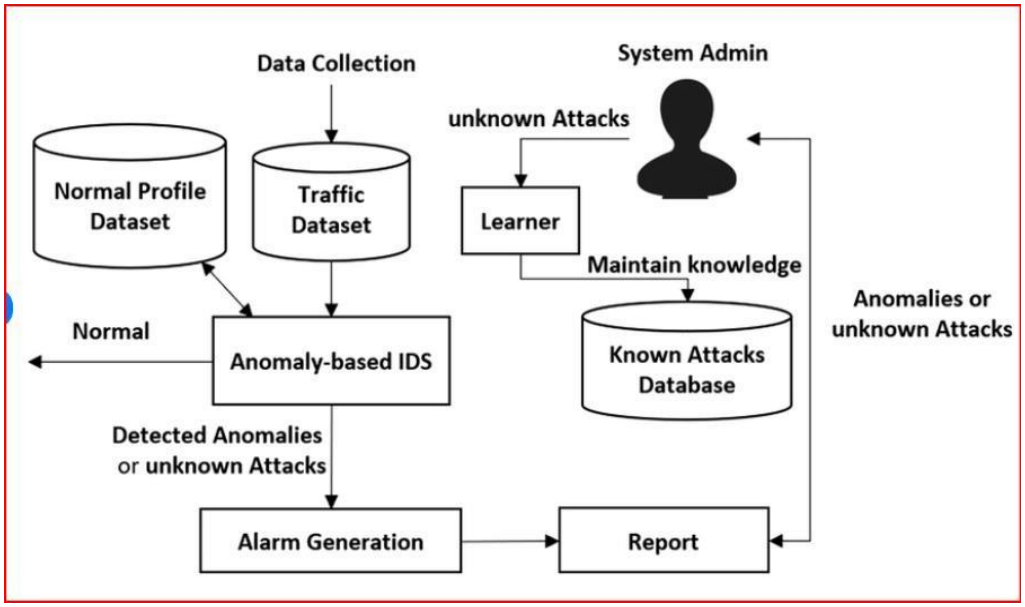

Figure 1. 5: IDS basé sur les anomalies [29]

6.2.3. IDS Réseau

Un IDS basé sur le réseau est généralement déployé sur les principaux hôtes ou commutateurs. La majorité des IDS basés sur le réseau sont indépendants du système d'exploitation (OS); ainsi, ils peuvent être appliqués dans différents environnements de système d'exploitation. En outre, les IDS basés sur le réseau sont capables de détecter des types spécifiques d'attaques de protocole et de réseau [15].

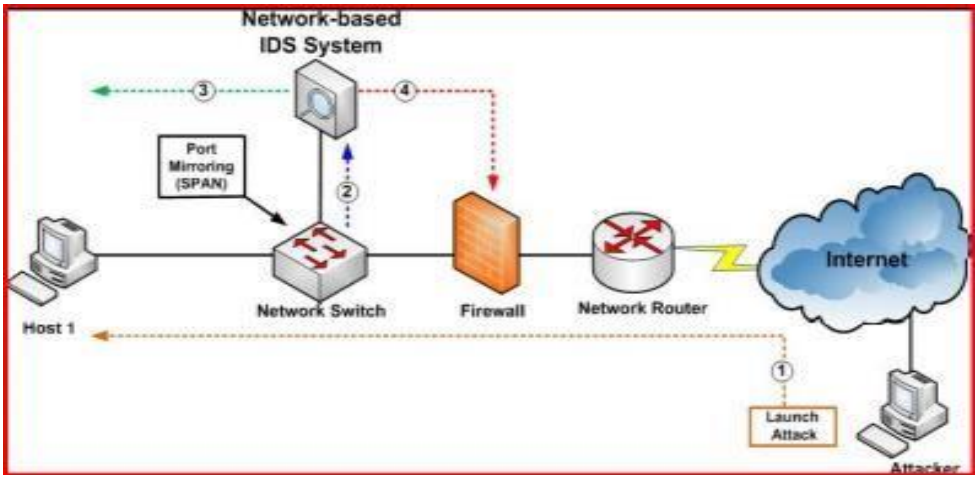

Figure 1. 6 : IDS Réseau [15]

6.2.4. IDS basés sur l'hôte

Ce type d'IDS analyse exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais "seulement" les activités d'un hôte, ils se montrent habituellement plus précis sur les types d'attaques. Ces IDS utilisent deux types de sources pour fournir une information sur l'activité : les logs et les traces d'audit du système d'exploitation [31].

6.2.5. IDS hybrides

Ces systèmes combinent les fonctionnalités d'un IDS réseau et d'un IDS hôte pour fournir une surveillance complète du système [32].

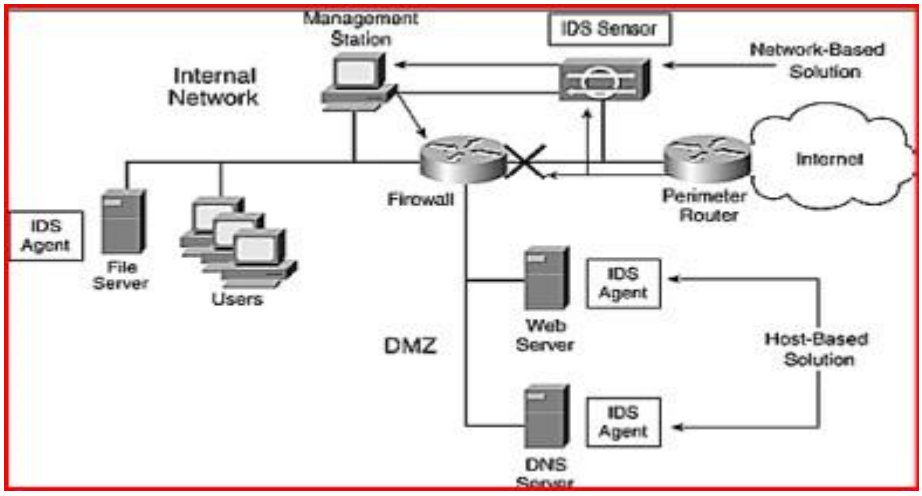

Figure 1. 7 : IDS hybride [15]

Conclusion

Dans ce chapitre, nous avons présenté une revue sur le cloud computing, le fog computing et la sécurité dans le fog et précisément les systèmes de détection d'intrusion. C'est une représentation du Fog computing Il est expliqué avec ses caractéristiques, ses avantages, son architecture, ses applications IoT, et ses problèmes de sécurité. Le Fog Computing peut être considéré comme une aide au cloud informatique non seulement pour fournir des services à la périphérie du réseau, mais aussi pour assurer la sécurité. Aussi, les attaques à la sécurité sont traitées dans ce chapitre. La discussion porte sur les techniques de sécurité. Le chapitre se termine en discutant sur les systèmes de détection d'intrusion.

Dans le chapitre suivant, nous allons détailler les différents concepts d'Intelligence Artificielle et d'une façon précise le deep learning et le federated learning, qui constituent l'objet de notre travail de recherche.

CHAPITRE 2 :

Federated Learning

Introduction

La nature nous montre tous les jours que l'intelligence ne se limite pas à l'être humain, ni même aux êtres vivants. Tout système qui peut s'adapter et répondre efficacement à son environnement peut être considéré comme intelligent, ce qui inclut la notion d'intelligence artificielle (IA).

L'intelligence artificielle (IA) est devenue un sujet d'une importance majeure dans notre époque, car est un domaine de l'informatique qui vise à créer des systèmes informatiques capables de réaliser des tâches normalement accomplies par des êtres humains.

Dans ce chapitre nous allons approfondir la notion d'IA, et sa relation avec le Machine Learning (ML) et le Deep Learning (DL). Ensuite nous aborderons les techniques de l'apprentissage, bien précisément l'apprentissage machine (machine Learning). Après on parlera plus en détails de l'apprentissage profond (Deep Learning), ou nous allons exposer la notion des réseaux de neurones artificiels. Ensuite nous allons présenter le principe du Federated Learning.

1. L'Intelligence Artificielle

1.1. Définition

Le terme "intelligence artificielle" a été créé par John McCarthy, un informaticien américain, considéré comme l'un des fondateurs de l'IA. L'IA, est une branche de l'IT qui vise, à l'aide de machines, à reproduire l'intelligence humaine en l'imitant généralement, par exécution de tâches. L'IA donc est un domaine très vaste qui englobe de nombreuses techniques différentes [33].

Il est difficile de définir exactement l'intelligence artificielle, mais il est important que la machine donne l'impression d'être intelligente en résolvant des problèmes, par exemple en imitant le comportement humain ou en utilisant des stratégies plus flexibles que celles de la programmation classique [33].

1.2. L’impact de l'Intelligence Artificielle

L'Intelligence Artificielle (IA) a envahi progressivement notre quotidien : nous posons des questions à Siri, nous utilisons Alexa pour contrôler notre maison à distance et nous nous ébahissons de la victoire d'AlphaGo face au champion du monde [34].

L'Intelligence Artificielle permet de réaliser des tâches plus rapidement avec plus de performances. Certains métiers pourront ainsi être automatisés et ne nécessiteront plus d’action humaine. Selon un récent rapport du Conseil français de l'emploi en France de janvier 2017, environ 10% des emplois disparaîtront et 50% seront transformés.

En exploitant l’Intelligence Artificielle de manière pertinente et en anticipant l'éducation liée à cette nouvelle situation, les emplois ne devraient pas être menacés. Prenons l'exemple du médecin. Si la machine est plus performante que lui pour repérer une maladie, l’expert, en l’occurrence le médecin, sera toujours plus qualifié dans la relation au patient. Si quelqu'un a un cancer, le patient préférera l'apprendre par une personne empathique plutôt que par une machine. Finalement, l'IA ne remplacerait pas nos métiers, mais y apporterait une valeur ajoutée [34].

1.3. La relation entre l’IA, ML et DL

L’apprentissage automatique (Machine Learning) et l’apprentissage profond (Deep Learning) font partie de l’intelligence artificielle. Alors que l’apprentissage profond est une branche de l’apprentissage automatique [35], comme illustré dans la figure 2.1.

Si le ML et le DL sont des intelligences artificielles, l’inverse n’est pas vrai. Par exemple, les tableaux de connaissances ou les moteurs de règles sont des intelligences artificielles, mais ne font pas partie du ML ou du DL [35].

Les algorithmes d'intelligence artificielle peuvent être utilisés pour la prise de décision et l'automatisation optimisée, par le biais du ML et le DL, pour augmenter l'efficacité d'une entreprise. En effet, l'évolution de l'IA, notamment ces dernieres années, a été considérable grâce à l'émergence du Cloud Computing et du Big Data, avec une puissance de calcul peu coûteuse et un accès possible à une grande quantité de données. Donc, les agents intelligents ne sont donc plus programmés, mais elles apprennent.[35].

Au final, le terme d’Intelligence Artificielle est souvent employé pour désigner le Machine Learning et le Deep Learning. En réalité, il désigne la capacité d’une machine à apprendre des concepts de manière

autonome, ce qui est une véritable révolution technologique qu'ont touché quasiment tous les domaines [35].

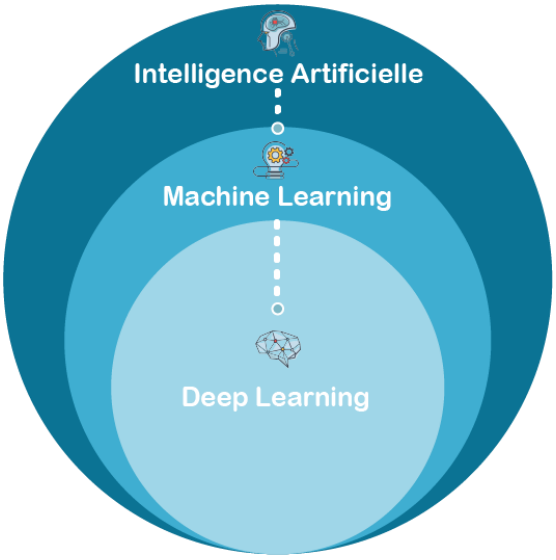

Figure 2. 1: Intelligence artificielle, machine learning et deep learning[36]

2. Machine Learning

En 1959, Arthur Samuel, un pionnier du domaine de l'apprentissage automatique, a fait l'énoncé suivant : « L'apprentissage automatique est le domaine d'étude qui fournit aux ordinateurs la capacité d'apprendre sans être explicitement programmés pour le faire » [37].

2.1. Définition

L'apprentissage automatique est un domaine qui se concentre sur l'analyse de données et la reconnaissance de motifs dans des ensembles de données. Son objectif est de développer des algorithmes capables d'apprendre à partir de ces données pour effectuer des prédictions. Pour ce faire, il est nécessaire de construire un modèle spécifique adapté à la nature des données et aux prédictions souhaitées. Il existe une grande variété de méthodes d'apprentissage automatique, chacune ayant ses propres caractéristiques uniques. Le choix de la méthode à utiliser dépendra du type de problème que l'on cherche à résoudre. Les données utilisées pour l'apprentissage peuvent être unidimensionnelles ou multidimensionnelles, avec des valeurs appelées caractéristiques ou attributs [38].

2.2. Types d'apprentissage automatique

Il existe plusieurs types d'apprentissage automatique (ou machine learning) dont les principaux sont :

2.2.1. Apprentissage supervisé

Les algorithmes d'apprentissage supervisé utilisant des données d'entrainement composées d'objets d'entrée, tels que des vecteurs, et des valeurs de sortie correspondantes. L'objectif est de trouver une fonction qui peut mapper les données d'entrée aux sorties souhaitées, afin de pouvoir prédire les sorties pour de nouvelles données d'entrée. Les algorithmes cherchent à minimiser à la fois le biais et l'erreur de variance des résultats prédits. Le biais est causé par des hypothèses simplificatrices de l'algorithme pour faciliter l'apprentissage de la fonction cible, mais un biais élevé peut entraîner des performances prédictives inférieures pour des problèmes qui ne répondent pas pleinement aux hypothèses [39].

2.2.2. Apprentissage non supervisé

Dans l'apprentissage non supervisé, la machine reçoit simplement les entrées, mais n'obtient ni sorties cibles supervisées, ni récompenses de son environnement. Il peut sembler un peu mystérieux d'imaginer ce que la machine pourrait éventuellement apprendre étant donné qu'elle ne reçoit aucun retour de son environnement. Cependant, il est possible de développer un cadre formel pour l'apprentissage non supervisé basé sur la notion que l'objectif est de construire des représentations de l'entrée qui peuvent être utilisées pour la prise de décision, la prédiction des entrées futures, communiquer efficacement les entrées à une autre machine, etc. [40]

2.2.3. Apprentissage par renforcement

L'apprentissage par renforcement (RL) est une classe de modèle d'apprentissage automatique où le processus d'apprentissage est basé sur les rétroactions évaluatives sans aucun signal supervisé [1, 8]. RL vise à créer des agents semblables aux humains, qui apprennent pour eux-mêmes par essais et erreurs, uniquement à partir de récompenses ou punitions, pour développer des stratégies efficaces qui finiront par conduire aux plus grandes récompenses à long terme [39].

2.3. Processus de l'apprentissage automatique

Pour développer un modèle de Machine Learning il faut passer par quatre étapes principales :

Étape 1 : Sélectionner et préparer l'ensemble de données d'entraînement

Ces données seront utilisées pour alimenter le modèle de Machine Learning pour apprendre à résoudre un problème spécifique. Les données peuvent être étiquetées, afin d'indiquer au modèle les caractéristiques qu'il devra identifier. Elles peuvent aussi être non étiquetées, et le modèle devra repérer et extraire les

caractéristiques récurrentes de lui-même. Ces données doivent être soigneusement préparées, organisées et nettoyées sinon l'entraînement du modèle de Machine Learning risque d'être biaisé [41].

Étape 2 : Sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement: Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre [42].

Étape 3 : Entraînement de l'algorithme: Il s'agit d'un processus itératif, ou des paramètres sont utilisés à travers l'algorithme, et les résultats sont comparés avec ceux qu'il aurait dû produire. Les poids et le biais peuvent ensuite être ajustés pour améliorer la précision des résultats [43].

Étape 4 : Utilisation et amélioration du modèle: On utilise le modèle sur de nouvelles données qui proviennent du problème à résoudre. Exemple : un modèle de Machine Learning conçu pour détecter des spams sera utilisé sur des emails [41]

2.4. IDS basés sur le Machine Learning

Les IDS basés sur la machine learning sont des systèmes de détection d'intrusion qui utilisent des algorithmes de Machine Learning pour analyser les données de trafic réseau et identifier les anomalies. Les IDS utilisent une variété de techniques, y compris l'apprentissage automatique (Machine Learning) et l'apprentissage en profondeur (Deep Learning), pour prévoir et découvrir les attaques avant qu'elles arrivent [44].

Les IDS basés sur le Machine Learning sont capables de détecter même les attaques inconnues. Néanmoins, le défi fondamental dans cette direction implique la conception d'une machine efficace IDS basée sur l'apprentissage qui fonctionne bien sur les données en temps réel [45].

L'invention concerne des procédés de système de détection d'intrusion basés sur l'apprentissage supervisé qui détectent des intrusions à l'aide de données d'apprentissage étiquetées. Dans l'apprentissage supervisé, les données sont organisées par paires, mappant une source de données hôte ou un réseau et une valeur de sortie connectée (c'est-à-dire une étiquette) qui doit être en mode normal, intrusion ou précis. Les résultats étiquetés sont pré-arrangés et utilisés pour enseigner à l'algorithme comment atteindre les résultats nécessaires pour les types de données non observés. Dans la méthode d'apprentissage non supervisé, les IDS peuvent détecter les intrusions en appliquant des données non étiquetées pour former le modèle [45].

3. Le Deep Learning

3.1. Définition

L'apprentissage profond « deep learning » est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années, à la différence que les algorithmes DL peuvent automatiquement apprendre des représentations à partir de données telles que des images, des vidéos ou des textes, sans introduire la connaissance du domaine humain. Le mot "profond" dans l'apprentissage profond représente les nombreuses couches d'algorithmes, ou réseaux neuronaux, qui sont utilisés pour reconnaître les modèles dans les données. Les architectures très flexibles de DL peuvent apprendre directement des données brutes, de la même façon que le cerveau humain fonctionne, et peuvent augmenter leur précision prédictive lorsqu'elles sont fournies avec plus de données [46].

En outre, l'apprentissage profond est la technologie principale qui permet une grande précision et la précision dans des tâches telles que la reconnaissance vocale, la traduction du langage et la détection d'objets. Il a mené à de nombreuses percées récentes dans l'IA, y compris AlphaGo de Google DeepMind, voitures autonomes, assistants vocaux intelligents, et bien d'autres [46].

3.2. Définition de réseaux de neurones

L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », est un système informatique inspiré du réseau neuronal biologique qui constitue le cerveau animal. De tels systèmes « apprennent » à effectuer des tâches en prenant en considération des exemples généralement sans être programmés avec des règles spécifiques à la tâche.

Le réseau neuronal artificiel utilise des modèles aux composants fortement connectés, composé de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples pour l'objectif d'améliorer les capacités de l'informatique. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite [47][48].

3.3. Fonctionnement de réseaux de neurones

L'architecture des réseaux profonds est organisée en plusieurs couches de neurones connectées pour n'importe quel type de ces réseaux. Le réseau neuronal est construit à partir de 3 types de couches [47] :

Couche d'entrée (Input Layer) : données initiales pour le réseau neuronal.

Couches cachées (Hidden Layers) : couche intermédiaire entre la couche d'entrée et la couche de sortie et endroit où tout le calcul est effectué. Chacune des couches cachées est constituée de plusieurs neurones, dans lesquels la force du signal d'un neurone dépend de facteurs tels que le biais, le poids et la fonction d'activation.

Couche de sortie (Output Layer) : produit le résultat pour des entrées données.

Chaque couche est composée d'un ou plusieurs nœuds, représentés dans la figure 2.5 par les petits cercles. Chaque paire de couches voisines est connectée. Les connexions entre eux appelées poids (Weights). Les "neurones" d'une même couche généralement appelés "noeuds" n'ont aucune association. La figure 2.5 illustre une architecture standard d'un modèle de réseau de neurones profond, dans ce type particulier de réseau de neurones, les informations ne circulent que de l'entrée à la sortie.

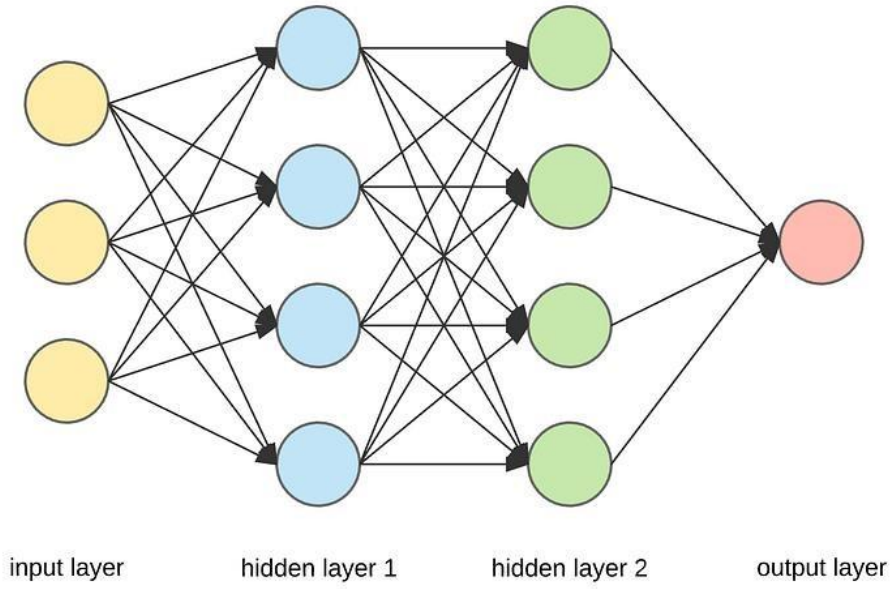

Figure 2. 2 : L'architecture de réseaux de neurones [47]

3.4. Types des réseaux de neurones

Il existe différents types de réseaux de neurones, et ils sont classés en fonction du nombre de nœuds cachées du modèle ou encore du nombre d'entrées et de sorties de chaque nœud.

La propagation des informations entre les différents neurones peut varier et dépend du type de réseaux de neurones.

3.4.1. Les réseaux de neurones dit "feed-forward" (à propagation avant)

C'est la variante la plus simple, l'information ne se déplace que dans une seule direction, elle traverse directement l'entrée aux nœuds de traitement (couches cachées) puis aux sorties, avec absence de cycle ou de boucle dans le réseau [54].

3.4.2. Les réseaux de neurones récurrents "feed-back"

Un réseau neuronal artificiel avec une structure d'information séquentielle est connu comme un réseau neuronal récurrent (RNN). Ils sont appelés récurrents parce qu'ils exécutent la même fonction sur chaque élément de séquence, avec le résultat en fonction des calculs antérieurs. Les RNNs sont des réseaux en boucle qui permettent la conservation des données [48].

A cet effet, dans un RNN, la sortie des neurones dans chaque couche est connectée à l'entrée des neurones de l'autre couche et aussi à elle-même. En outre, dans les RNNs, la couche d'entrée est unidirectionnellement connectée aux couches cachées, tandis que les neurones des couches cachées sont connectés à eux-mêmes et à tous les autres neurones de la couche suivante pour un échange d'informations complet [50], la structure d'un RNN est illustrée à la figure 2.2.

En ce qui concerne les corrélations temporelles des attaques de sécurité et des comportements malveillants, les RNNs peuvent être efficacement utilisés pour les modéliser. À cette fin, les RNNs peuvent être formés en utilisant les entrées actuelles et historiques, dans lequel la probabilité d'une attaque est basée sur les états actuels et antérieurs des fonctionnalités (features). [50]

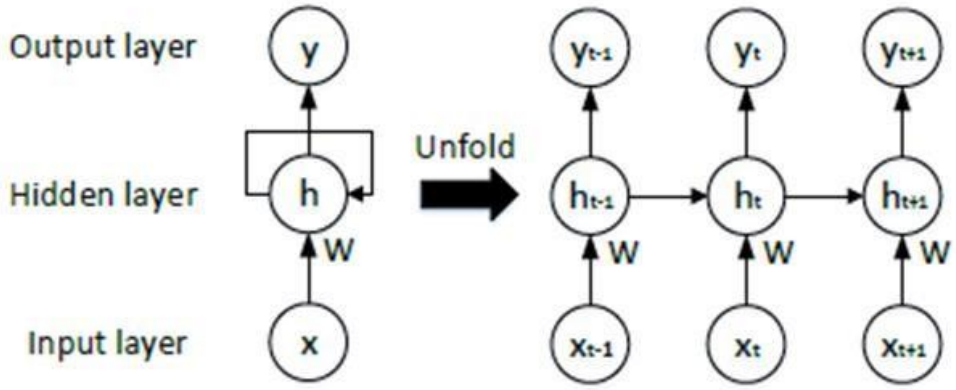

Figure 2. 3: La structure de RNN [51].

3.5. LSTM (Long Short-Term Memory)

Le réseau RNN a un long pas de temps car il prend en compte l'état sauvegardé précédent lors de la mise à jour du poids, les gradients lorsque l'entraînement devient de plus en plus petit et après quelques étapes, les erreurs n'ont pas pu être propagées à la fin du réseau. Il n'y aura pas de différence significative dans le résultat, donc il ne peut pas faire de mise à jour des poids. Pour surmonter ce problème, une architecture à mémoire longue et courte durée (LSTM) a été proposée. L'idée clé de la méthode LSTM est la capacité de supprimer ou d'ajouter des informations à l'état de la cellule. Cette technique est réglée par des structures appelées portes (Gates) [52].

La fonction sigmoïde est utilisée pour contrôler les portes de la cellule LSTM, qui déterminent le flux d'information à garder, oublier ou ajouter à la mémoire à court terme. Elle est généralement appliquée aux " portes " (gates) de l'unité LSTM, où une valeur de 1 signifie que toutes les informations passent et une valeur de 0 signifie le contraire [52].

Ainsi la fonction tangente hyperbolique (tanh) est utilisée pour calculer l'état de la cellule à court terme (cell state) dans un LSTM. Elle permet de réguler les valeurs de l'état de la cellule en les comprimant entre -1 et 1, ce qui permet de contrôler le flux d'information à travers les portes [52].

Les cellules LSTM sont les plus efficaces pour retenir les informations utiles lors de la rétro-propagation du gradient. Ce qui a leur permet de corriger les différences entre les prédictions sortantes et les catégories de référence en calculant le gradient de l'erreur pour chaque neurone, en allant de la dernière couche vers la première [52].

3.5.1. Architecture des LSTM

Une cellule d'un réseau LSTM est principalement composée d'un input gate, un output gate et un forget gate, comme il est illustré à la figure 2.3.

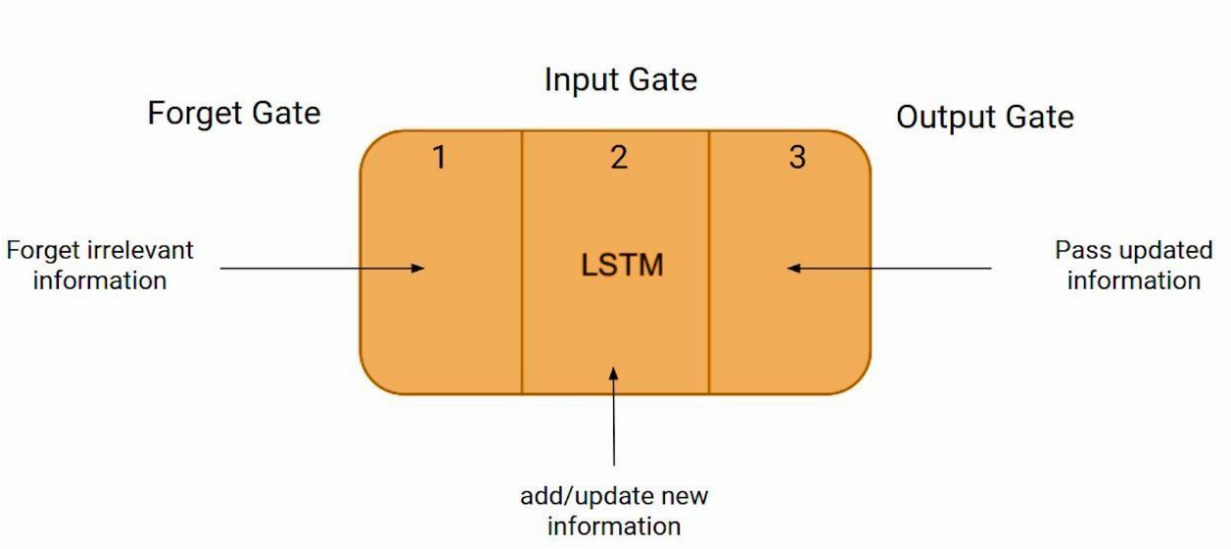

Figure 2. 4 : Schéma simplifié d'une cellule d'un réseau LSTM [53].

La principale idée derrière un LSTM est de diviser le signal qui traverse notre réseau en deux parties bien distinctes [53] :

- Court Terme à travers le hidden state.
- Long Terme à travers le cell state.

3.5.2. Les Étapes LSTM

Un réseau LSTM effectué durant chaque passage les 5 étapes [53] suivantes :

- Détection des informations passées dans le cell state via le forget gate.
- Choix des informations pertinentes à long terme à travers l'input gate.
- Ajout des informations choisies au cell state.
- Détection des informations importantes à court terme dans le cell state.
- Génération du nouveau hidden state à travers l'output gate.

La relation de récurrence d'un LSTM comprend donc une variable h pour le hidden state et une variable c pour le cell state : $h_t , c_t = f(x_t , h_{t-1} , c_{t-1})$.

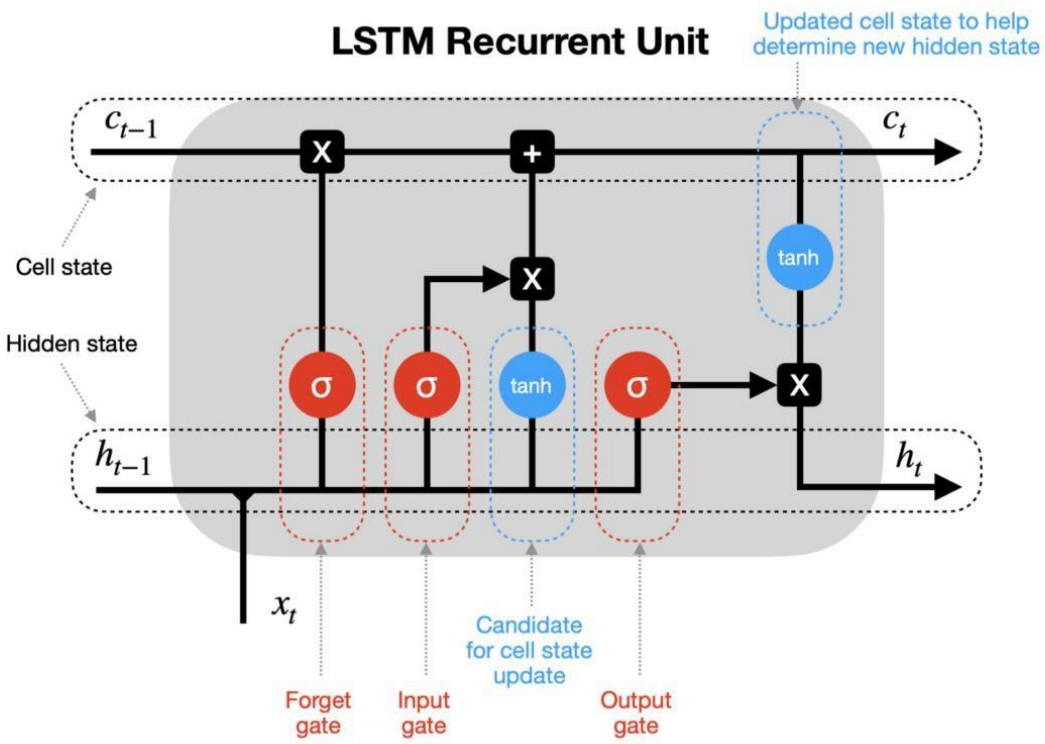

Figure 2. 5 : Cellule d'un réseau LSTM [53].

3.6. Fonction d'activation

Une fonction d'activation, ou de transfert, permet de changer notre manière de voir une donnée. Autrement dit c'est une fonction mathématique utilisée sur un signal. Elle applique une transformation sur des données d'entrée pondérées (multiplication matricielle entre les données d'entrée et les poids). La fonction peut être linéaire ou non linéaire.

Les fonctionnalités (features) de 1000 couches de transformations linéaires pures peuvent être reproduites par une seule couche (car une chaîne de multiplication matricielle peut toujours être représentée par une seule multiplication matricielle). Une transformation non linéaire peut toutefois créer de nouvelles relations de plus en plus complexes. Ces fonctions sont donc très importantes dans le deep learning, pour créer des fonctionnalités de plus en plus complexes avec chaque couche. [54]

Un modèle étant composé de multiples couches, et donc de multiples fonctions d'activation, des changements successifs et complexes de représentation s'opèrent. Cela permet d'avoir un nouveau point de vue sur nos données que l'homme serait incapable d'avoir en peu de temps. Chaque neurone d'une couche

va appliquer la fonction d'activation de la couche sur les données. Cette transformation sera différente selon chaque neurone car chacun possède un poids différent.

Pour choisir la bonne fonction d'activation il faut à la fois considérer la transformation directe qu'elle applique aux données mais aussi son dérivé qui sera utilisée pour ajuster les poids lors de la back propagation[54]. Voici quelle que fonction les plus utilisées [54] :

3.6.1. Sigmoid

La fonction Sigmoïde est l'une des premières fonctions utilisées par les experts du domaine qui offre en plus l'avantage de normaliser les entrées qu'elle reçoit. Elle donne une valeur entre 0 et 1, une probabilité, donc elle est très utilisée pour les classifications binaires, lorsqu'un modèle doit déterminer seulement deux labels. Ainsi, pour la classification des critiques de cinéma, plus la valeur retournée par Sigmoïde est proche de 1 plus le modèle en considère que la critique est positive. Au contraire, plus elle est proche de 0, plus elle est considérée comme négative [55] [56].

fonction_Sigmoid(x) = 1 / (1 + exp(-x))

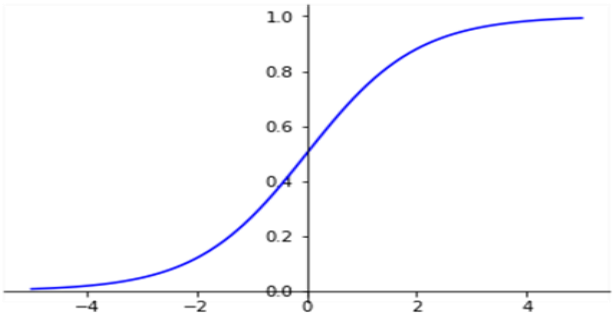

Figure 2. 6 : Fonction Sigmoïde [56]

3.6.2. tanh

La fonction tanh est simplement la fonction de la tangente hyperbolique.

Il s'agit en fait d'une version mathématiquement décalée de la fonction sigmoïde. sigmoïde donne un résultat entre 0 et 1, et tanh donne un résultat entre -1 et 1.

L'avantage de tanh est que les entrées négatives seront bien répertoriées comme négatives là où, avec sigmoïde, les entrées négatives peuvent être confondus avec les valeurs proches de nulles. donc Tanh fonctionne mieux que la fonction sigmoïde dans la plupart des cas.

Cette fonction est, comme Sigmoïde, utilisée dans la classification binaire.[56]

fonction_tanh(x) = sinh(x)/cosh(x) fonction_tanh(x) = ((exp(x) – exp(-x))/(exp(x) + exp(-x)))

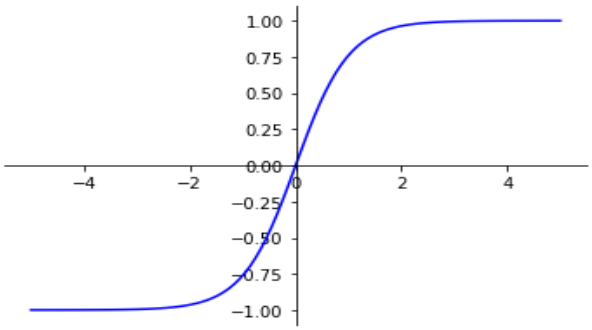

Figure 2. 7: Fonction tanh [56].

3.6.3. Softmax

La fonction Softmax permet de transformer un vecteur réel en vecteur de probabilité.

On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multiclasse. Dans la fonction Softmax, chaque vecteur est traité indépendamment.

L'argument axis définit l'axe d'entrée sur lequel la fonction est appliquée [56].

$$\text{fonction\_Softmax}(x) = \exp(x) / \text{sum}(\exp(x_i))$$

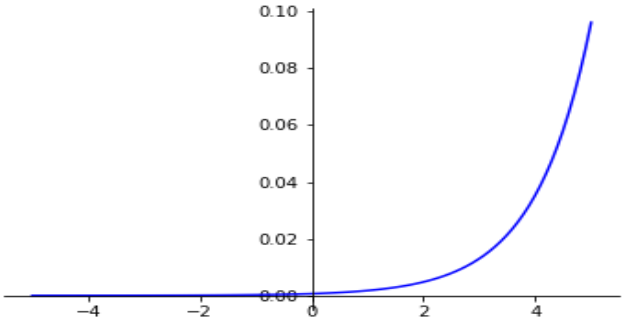

Figure 2. 8: Fonction Softmax [56].

3.7. Fonction de perte (loss function)

Les fonctions de perte sont la mesure du rendement du réseau neuronal. En termes simples, la fonction de perte est nécessaire pour calculer le déficit entre les valeurs réelles et les valeurs prévues par notre réseau. Plus la valeur de la perte est faible, plus la performance de notre réseau est bonne et plus la valeur est élevée, plus nos valeurs prévues sont éloignées des valeurs réelles. Et enfin, notre objectif est de minimiser cette perte [57].

3.7.1. La binary cross-entropy

Est une mesure de perte utilisée lorsque les résultats catégoriques sont binaires, ce qui signifie qu'ils peuvent prendre deux valeurs possibles : oui/non. Cette méthode est couramment employée dans les modèles de régression logistique [58].

3.7.2. La categorical cross-entropy

Est une mesure de perte utilisée lorsque les résultats de la catégorie ne sont pas binaires, ce qui signifie qu'il existe plus de deux valeurs possibles, par exemple : Type 1/Type 2/.../Type n." [58].

3.8. Optimiseur

Une fois la fonction de perte définie, un optimiseur est utilisé pour ajuster les paramètres du modèle afin de minimiser la fonction de perte. Il convient également de mentionner que ces optimisateurs peuvent être ajustés avec différents paramètres ou hyperparamètres tels que le taux d'apprentissage, le momentum, le taux de décroissance, etc.

En outre, ces optimisateurs peuvent être combinés avec différentes techniques telles que la planification des taux d'apprentissage, ce qui peut aider à améliorer les performances du modèle [59].

3.8.1. Algorithme d'optimisation Adam

Qui signifie « Adaptive Moment Estimation », est le plus utilisé dans le domaine de l'apprentissage profond. Cette technique d'optimisation calcule un taux d'apprentissage adaptatif pour chaque paramètre. Elle définit la dynamique et la variance du gradient de la perte et exploite un effet combiné pour mettre à jour les paramètres de poids [60] [61].

3.9. Cross validation

La validation croisée est une méthode statistique d'évaluation et de comparaison des algorithmes d'apprentissage en divisant les données en deux segments : l'un utilisé pour apprendre ou former un modèle (train) et l'autre utilisé pour valider le modèle. Dans la validation croisée typique, les ensembles de formation et de validation doivent être croisés en cycles successifs de sorte que chaque point de données a une chance d'être validé par rapport. La forme de base de la validation croisée est k-fold cross-validation. D'autres formes de validation croisée sont des cas particuliers de validation croisée k-fold ou impliquent des cycles répétés de validation croisée k-fold.

Dans la validation croisée de k-fold, les données sont d'abord divisées en k segments ou plis de taille égale (ou presque égale). Par la suite, k itérations de formation et de validation sont effectuées de telle sorte que, dans chaque itération, une partie différente des données est présentée pour validation, tandis que les autres k 1 plis sont utilisés pour l'apprentissage [62].

3.10. Early stopping

Un problème dans les réseaux neuronaux est de choisir le nombre d'époques pendant la formation, trop d'époques dépasseront le modèle, alors que trop moins pourrait causer un sous-ajustement. Early Stopping est un callback utilisé lors de la formation des réseaux neuronaux, ce qui nous donne l'avantage d'utiliser un grand nombre d'époques de formation et d'arrêter la formation une fois que les performances du modèle cessent d'améliorer sur l'ensemble de données de validation [63].

3.11. La détection d'intrusion basée sur le Deep learning

Les méthodes de détection basées sur le deep Learning apprennent automatiquement la fonction. Ces types de méthodes fonctionnent de bout en bout et deviennent progressivement l'approche dominante dans les études des IDS.

Plusieurs approches de deep learning ont été étudiées récemment pour la détection d'intrusion. Dans les systèmes de détection basé sur la signature, les intrusions sont détectées en comparant les comportements surveillés avec des motifs d'intrusion prédéfinis, tandis que les systèmes basés sur les anomalies se concentrent sur la connaissance du comportement normal afin d'identifier toute déviation et toutes activités suspectes. Les méthodes de deep learning sont applicables pour les 2 types de détection grâce à ces capacités qui permettent d'extraire des niveaux plus élevés de relations non linéaires entre les données, afin d'identifier toute déviation d'une activité bénigne. Cependant, ces méthodes exigent une quantité énorme des données dans le but de détecter et identifier les motifs de différentes classes [64].

4. Federated Learning

Dans cette section, nous allons présenter la notion d'apprentissage fédéré, nommé communément « federated learning ».

4.1. Définition

L'apprentissage fédéré (Federated Learning) est une approche d'apprentissage automatique distribué qui permet la formation sur un large corpus de données décentralisées résidant sur des appareils comme les téléphones portables. Federated Learning est un exemple du plus général approche consistant à "apporter le code aux données, au lieu de données au code » et aborde les problèmes fondamentaux de la confidentialité, de la propriété et de la localité des données [65].

4.2. Fonctionnalité

Cela fonctionne comme ceci : votre appareil télécharge le modèle actuel, l'améliore en apprenant à partir des données de votre téléphone, puis résume les modifications sous la forme d'une petite mise à jour ciblée. Seule cette mise à jour du modèle est envoyée au cloud/fog, à l'aide d'une communication cryptée, où elle est immédiatement moyennée avec d'autres mises à jour d'utilisateurs pour améliorer le modèle partagé. Toutes les données d'entraînement restent sur votre appareil et aucune mise à jour individuelle n'est stockée dans le cloud [65].

4.3. Types

4.3.1. Apprentissage fédéré horizontalement (HFL)

L'apprentissage fédéré horizontal est utilisé dans les cas où chaque appareil contient un ensemble de données avec le même espace de fonctions mais avec des exemples d'instances différents [66].

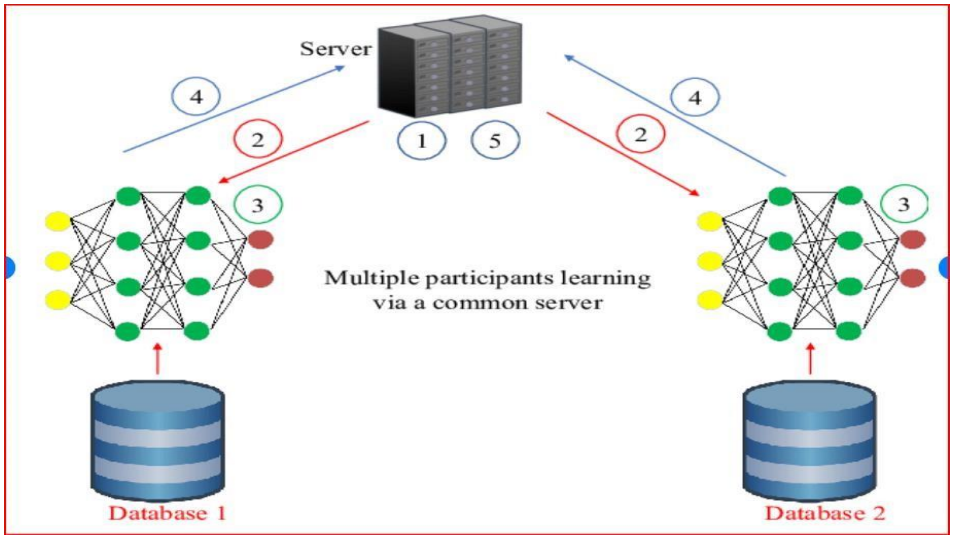

Figure 2. 9: Federated Learning System (Horizontal) [67]

4.3.2. Apprentissage fédéré verticalement (VFL)

VFL est appelé apprentissage fédéré basé sur les fonctionnalités, qui est adapté aux scénarios que les ensembles de données appartenant à différentes parties partagent le même espace d'identification d'échantillon mais diffèrent dans l'espace de caractéristiques. [66]

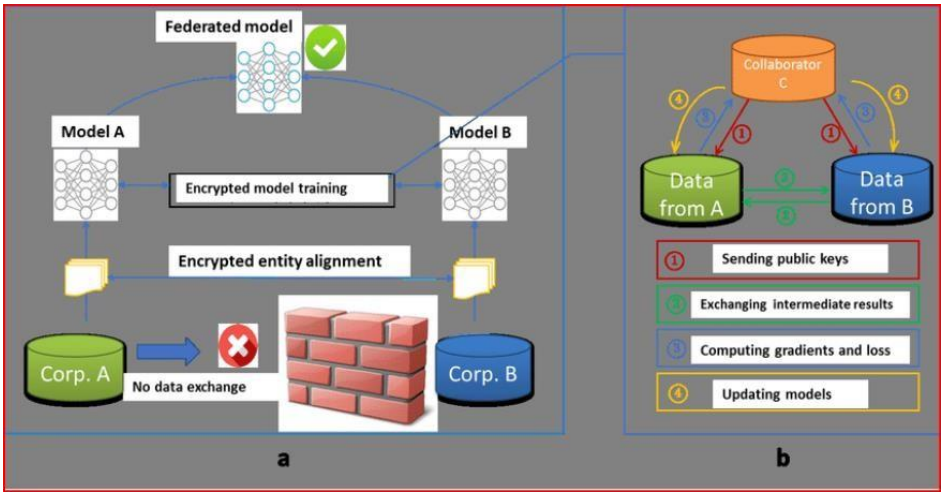

Figure 2. 10: Architecture for a vertical federated learning system. [68]

4.3.3. Apprentissage fédéré par transfert (FTL)

L'apprentissage par transfert vise à transférer des connaissances de domaines existants vers un nouveau domaine. Dans le cadre du transfert à l'apprentissage, les domaines sont souvent différents mais liés, ce qui rend possible le transfert de connaissances. L'idée clé est de réduire la divergence de distribution entre les différents domaines [69].

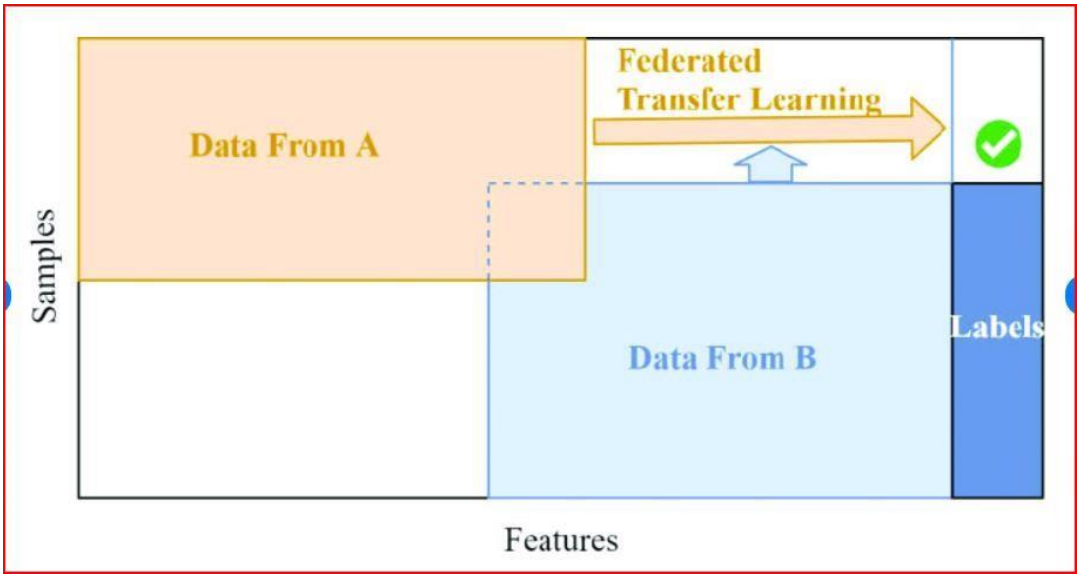

Figure 2. 11: Federated Transfer Learning (FTL) architecture. [70]

4.4. Application

L'apprentissage fédéré s'applique mieux dans les situations où les données de l'appareil sont plus pertinentes que les données qui existent sur serveurs (par exemple, les appareils génèrent les données en premier lieu), est sensible à la vie privée, ou autrement indésirable ou impossible à transmettre aux serveurs. Les applications actuelles de l'apprentissage fédéré concernent les tâches d'apprentissage supervisé, généralement à l'aide d'étiquettes déduit de l'activité de l'utilisateur [71].

4.6. L'approche Centralisée

L'apprentissage fédéré centralisé est une approche hybride de l'apprentissage machine qui combine des éléments de l'apprentissage centralisé et de l'apprentissage fédéré. Dans cette approche, certains modèles sont formés de manière centralisée en utilisant toutes les données disponibles, tandis que d'autres modèles sont formés de manière distribuée en utilisant des données locales sur des appareils clients. L'approche federated-centralized est particulièrement utile dans les scénarios où les données sont très hétérogènes ou certaines données sont trop sensibles pour être partagées entre les clients. Dans ces cas, il peut être nécessaire de former des modèles centraux à partir des données agrégées, tout en utilisant des appareils clients pour former des modèles locaux sur des sous-ensembles de données [72].

4.7. IDS basé sur l'apprentissage fédéré

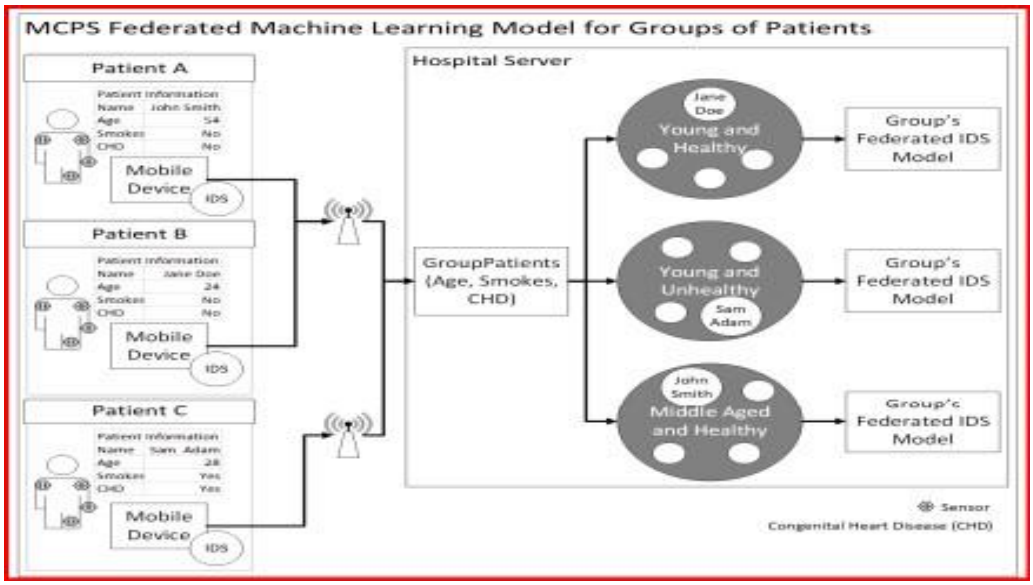

Figure 2. 12: Federated Learning Based IDS (FLIDS) [73]

Un modèle d'apprentissage fédéré (FL) a été conçu pour améliorer la détection d'anomalies distribuées près des sources d'anomalies/attaques. L'utilisation du calcul distribué ou en périphérie permet de bloquer une zone infectée sans perturber le fonctionnement global du système, ce qui améliore l'efficacité de la production. De plus, FL permet aux sites périphériques de partager des informations de modèle entre eux pour optimiser la détection d'anomalies à l'échelle mondiale, résolvant ainsi le problème de manque de données d'entraînement dans chaque site périphérique, en particulier pour des ensembles de données multivariées et de grande dimension. En conséquence, le temps de réponse du système lors d'attaques peut être amélioré [74].

Conclusion

Nous nous sommes intéressés en premier lieu dans ce chapitre à la représentation de l'intelligence artificielle et la Machine Learning comme concept général. Ensuite nous avons approfondi l'un de ses types qui est l'apprentissage profond (Deep learning) et à ce niveau nous avons abordé la notion de réseaux de neurones, et enfin nous avons enchaîné avec le principe de federated learning. Dans le chapitre 3 nous allons focaliserons sur l'utilisation d'un système de détection d'intrusion (IDS) basé sur le deep learning et le federated learning pour la détection d'attaques. Pour cela, en utilisant un ensemble de données (dataset).

CHAPITRE 3 :

Modélisation d’un modèle d’apprentissage profond.

Introduction

L’architecture distribuée du fog computing présente des défis en termes de sécurité. La dispersion des données, la latence et la connectivité intermittente peuvent rendre les environnements de fog vulnérables aux intrusions et aux attaques malveillantes. C'est là qu'interviennent les IDS (Intrusion Detection Systems) basés sur le federated learning, offrant une solution prometteuse pour renforcer la sécurité du fog computing.

Dans ce chapitre, nous nous concentrerons sur la détection d'attaques en utilisant un système de détection d'intrusion (IDS) basé sur le deep learning et le federated learning, en utilisant un ensemble de données (dataset).

1. Problématique

Les environnements de fog computing présentent des défis uniques en matière de sécurité, tels que la latence, la connectivité intermittente et la distribution des données.

L'expansion rapide de l'Internet des objets (IdO) a révolutionné la connectivité des appareils, mais elle présente des limitations et des vulnérabilités. Les dispositifs IoT ont une puissance de calcul limitée, une capacité de stockage réduite et une autonomie de batterie courte, ce qui en fait des cibles pour les acteurs malveillants. Les attaquants exploitent ces vulnérabilités pour perturber le brouillard informatique (fog computing) et former des botnets capables de lancer des attaques coordonnées. Les attaques incluent le déni de service, la manipulation des données et la propagation de logiciels malveillants. La détection des cyberattaques dans les infrastructures IoT est donc essentielle, mais les systèmes traditionnels de détection d'intrusion rencontrent des difficultés. Ils reposent sur des centres de surveillance centralisés et posent des problèmes de confidentialité et de sécurité des données.

La question qui se pose est donc :

Comment garantir une détection efficace des intrusions dans les environnements de fog computing, tout en respectant les contraintes de confidentialité des données et en s'adaptant à la nature distribuée du fog computing ?

2. Solution proposée

En utilisant le federated learning, les IDS dans le fog peuvent bénéficier d'une approche décentralisée et collaborative pour l'entraînement des modèles. Cette solution permet aux nœuds de fog de conserver leurs données localement tout en contribuant aux connaissances globales du modèle, offrant ainsi une détection des intrusions efficace, adaptée à la nature distribuée du fog computing, tout en préservant la confidentialité des données.

Pour évaluer la performance des approches de détection d'anomalies dans les réseaux informatiques de brouillard (fog computing) , l'ensemble de données Bot-IoT sert de ressource précieuse. Cet ensemble de données intègre le trafic réseau IoT original et simulé, en intégrant une gamme de scénarios d'attaque.
L'analyse de l'ensemble de données Bot-IoT permet aux chercheurs d'évaluer l'efficacité de diverses méthodes de détection d'anomalies dans l'identification et l'atténuation des attaques réseau IoT. Cette évaluation aide à renforcer la sécurité des environnements informatiques de brouillard en identifiant les vulnérabilités et en améliorant les mécanismes de détection.

3. Objectifs de projet

L'objectif ultime de notre projet est de fournir un système de détection d'intrusion performant, précis et évolutif, capable de protéger et sécuriser les réseaux dans des environnements distribués et décentralisés tels que le Fog Computing. En exploitant les avantages de l'apprentissage fédéré, nous cherchons à offrir une solution qui peut s'adapter aux variations du réseau, maintenir la confidentialité des données et améliorer continuellement les capacités de détection d'intrusion grâce à l'apprentissage collaboratif entre les clients et le serveur central.

4. L'architecture de projet

Dans notre étude, nous avons mis en œuvre un système de détection d'intrusion qui exploite le concept de l'apprentissage fédéré dans un environnement de Fog Computing. Cela est illustré dans la figure 3.1 cidessous :

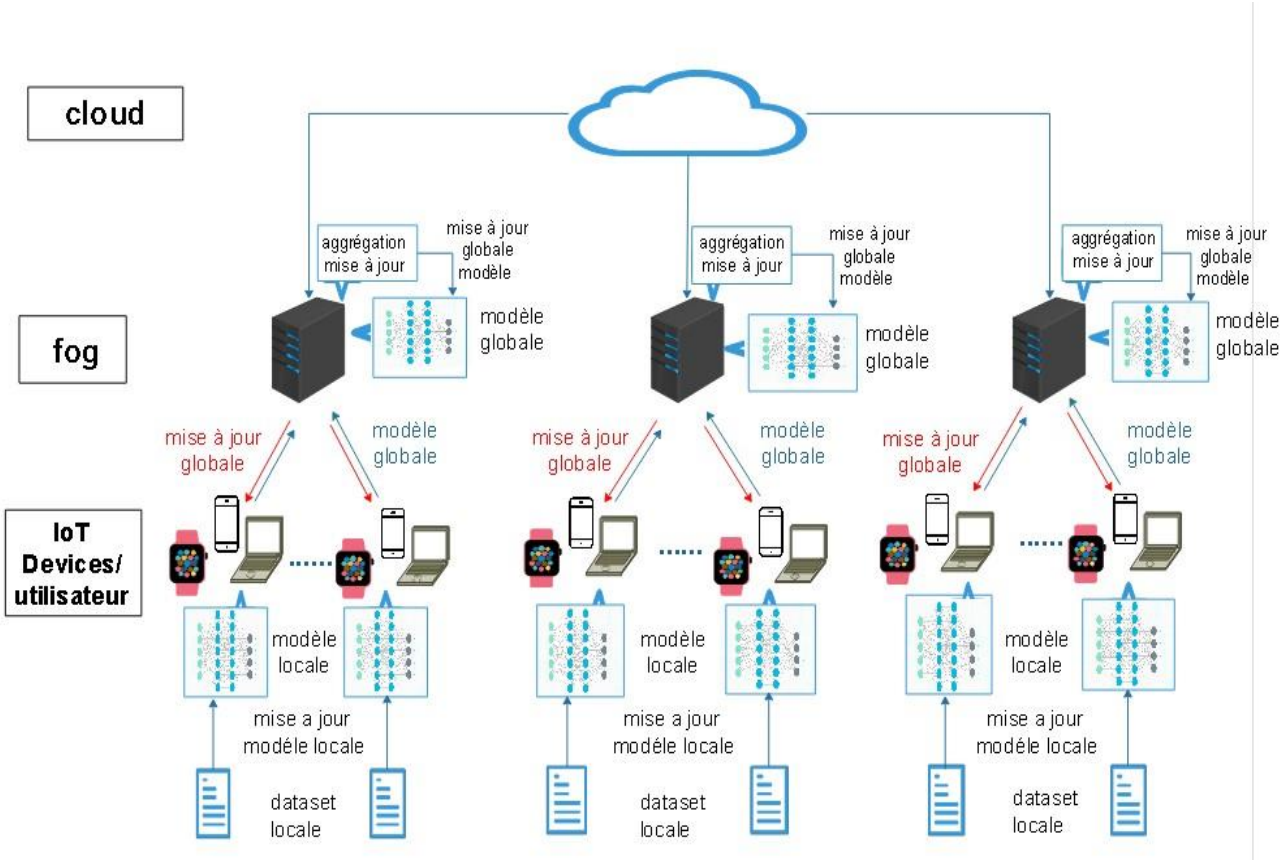

Figure 3. 1: architecture d'IDS basé sur federated learning dans le fog

L'apprentissage fédéré permet aux clients d'apprendre de manière partagée tout en conservant l'ensemble des données d'entraînement sur leurs machines et en dissociant la capacité d'apprentissage automatique de la nécessité de stocker les données côté serveur.

Le fonctionnement global de l'apprentissage fédéré est le suivant :

- Dans un serveur central, nous avons créé un modèle global qui servira de référence pour la détection d'intrusion.
- La machine télécharge le modèle global qui est situé au niveau du serveur et l'améliore en l'entraînant avec les données de la machine.
- Par la suite, la machine renvoie la mise à jour du modèle au serveur.
- Ainsi, le serveur effectue une agrégation avec d'autres mises à jour de clients pour améliorer le modèle global.
- Les clients n'ont alors plus besoin de divulguer leurs propres données personnelles et ces dernières demeurent sur leurs machines.

L’apprentissage fédéré est principalement composé de quatre grandes étapes :

Etape 1 : La première étape est de créer un modèle global dans le serveur fog.

Etape 2 : La seconde étape consiste à effectuer la diffusion du modèle vers plusieurs machines

Décentralisés.

Etape 3 : La troisième étape est celle au cours de laquelle les machines décentralisées entament leur entraînement sur le modèle diffusé par le serveur sur leurs propres données.

Etape 4 : La dernière étape consiste à récolter le résultat renvoyé par les différents modèles et représentant le gradient local puis à réaliser une agrégation de ces derniers pour produire un modèle plus performant.

Ce processus se répète plusieurs fois pour améliorer progressivement le modèle global en exploitant les connaissances locales de chaque client. Cela permet une détection d'intrusion distribuée et collaborative, où les clients contribuent à l'amélioration du modèle global en utilisant leurs propres données tout en préservant la confidentialité de ces données

5. Environnement de travail

5.1. Environnement matériel

Pour le développement de notre approche, nous avons utilisé un ordinateur ASUS possédant les caractéristiques suivantes :

- processeur : Intel(R) Core(TM) i3-5010U CPU @ 2.10GHz 2.10 GHz
- Mémoire : 6 GO
- Système d’exploitation : Windows 10 64 bits

5.2. Environnement d’exécution

5.2.1. Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l’apprentissage automatique. Cette plateforme permet d’entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d’installer quoi que ce soit sur notre ordinateur à l’exception d’un navigateur.

5.2.2. Langage utilisé 

Nous avons choisi Python comme langage de développement. Python est un langage de programmation interprété, multi-paradigme et multiplateforme. Il est largement utilisé dans le domaine de l'apprentissage automatique et de l'intelligence artificielle en raison de sa flexibilité et de sa popularité. De plus, Python bénéficie d'un vaste éventail de bibliothèques logicielles open source, ce qui facilite grandement le développement de projets dans ces domaines.

5.2.3. Bibliothèques et Frameworks utilisées

Nous utilisons plusieurs bibliothèques dans notre projet, telles que *Pandas*, *Numpy* et *Scikit-Learn*. Pandas et Numpy sont utilisés pour la manipulation des données, notamment le chargement, la réorganisation et le traitement des données. Quant à Scikit-Learn, il nous permet d'expérimenter rapidement et facilement différentes techniques et algorithmes d'apprentissage automatique et d'analyse de données prédéfinis

Nous avons choisi d'utiliser les frameworks *TensorFlow* et *Keras* pour implémenter les méthodes de deep learning proposées. TensorFlow est une bibliothèque open source de deep learning développée par Google. Elle est largement utilisée pour effectuer des opérations numériques complexes et modéliser des architectures de deep learning. TensorFlow permet également un déploiement facile des calculs sur différentes plates-formes, telles que les CPU et les GPU. Keras, quant à lui, est une interface haut niveau qui fonctionne avec TensorFlow, considéré comme une puissante de la bibliothèque Python, facile à utiliser pour développer et évaluer des modèles d'apprentissage en profondeur.

6. Présentation du dataset

Le Bot-IoT est un ensemble de données de l'Université de Nouvelle-Galles du Sud (UNSW) qui a été publié en 2018 au format csv. Cet ensemble de données est utilisé pour la recherche en sécurité des dispositifs Internet des objets (IoT) et contient des données de trafic réseau qui ont été recueillies à partir de dispositifs IoT malveillants. Les fichiers ont été séparés, en fonction de la catégorie d'attaque et de la sous-catégorie, pour mieux aider au processus d'étiquetage [75].

L'ensemble de données Bot IoT de 48 fonctionnalités est une extension de l'ensemble de données Bot-IoT original, cet ensemble peut être utilisé pour former des modèles de détection d'intrusion pour les dispositifs IoT, analyser le trafic réseau et mener des recherches en sécurité informatique. Contrairement à l'ensemble de données original, qui ne comportait que 17 fonctionnalités, l'ensemble de données Bot IoT de 48 fonctionnalités comprend 48 fonctionnalités qui décrivent les caractéristiques du trafic réseau, telles que le

nombre de paquets, la taille des paquets, les ports source et de destination, les fonctionnalités basées sur le temps, les valeurs statistiques, caractéristiques du débit, et plus encore [75].

Afin de réduire le nombre de fonctionnalités de l'ensemble de données IoT-Bot, il est important de passer par des étapes pour nettoyer notre dataset de 48 fonctionnalités à 10 fonctionnalités.

Notre dataset est partagé en deux parties, une d'étiquette (label) et autre de fonctionnalités (features). Les données de dataset set sont représentées dans le tableau 3.1.

Dans le contexte du dataset Bot IoT, le but des attaques DDoS (Distributed Denial of Service) et DoS (Denial of Service) est généralement lié à la prise de contrôle d'appareils Internet des objets (IoT) compromis, formant ainsi un botnet. Les attaquants utilisent ensuite ces botnets pour lancer des attaques DDoS ou DoS massives.

Les attaques de reconnaissance dans les ensembles de données bots IoT visent à identifier et à analyser les dispositifs IoT vulnérables. Ces attaques peuvent être effectuées par des acteurs malveillants afin de constituer une liste de cibles potentielles pour d'éventuelles exploitations ultérieures, telles que des attaques de botnet ou des intrusions dans les systèmes IoT. Concernant l'attaque "Theft", nous avons la supprimer pour éviter le déséquilibre des données.

	Données	Description
Fonctionnalités	Mean	Durée moyenne au niveau agrégé des enregistrements
	Min	Durée minimale au niveau de l'agrégat des enregistrements
	Stddev	Écart-type des enregistrements agrégés
	N_IN_Conn_P_SrcIP	Nombre de connexions entrantes par IP source
	state_number	Représentation numérique de l'état des caractéristiques
	Stare	Paquets source vers destination par seconde
	stare N_IN_Conn_P_DstIP	Nombre de connexions entrantes par IP de destination.
Etiquettes	Attack	Paquets source vers destination par étiquette de classe: 0 pour le trafic normal, 1 pour le trafic d'attaque en second
	Category	Catégorie de trafic
	Subcategory	Sous-catégorie de trafic subcategory

Tableau 3. 1 : Les données sélectionnées [76]

catégorie d'attaque		nombre d'instances	
Attack	DDoS	438553	966682
	DoS	375152	
	Reconnaissance	152977	
Normal	Normal	291315	291315

Tableau 3. 2: catégorie des attaques.

7. Méthodologie

Dans cette section, nous présentons la méthodologie du travail effectué.

7.1. Modélisation

Dans cette étape, nous allons examiner les méthodes employées pour l'analyse de notre ensemble de données composé de 19 caractéristiques. Nous allons effectuer le nettoyage et la préparation des données pour les adapter à nos algorithmes en éliminant et en ajustant certains attributs indésirables. L'implémentation d'une étape de prétraitement des données, comme indiqué à la Figure 3.2, permet d'obtenir une formation plus fiable et, par conséquent, un modèle plus précis.

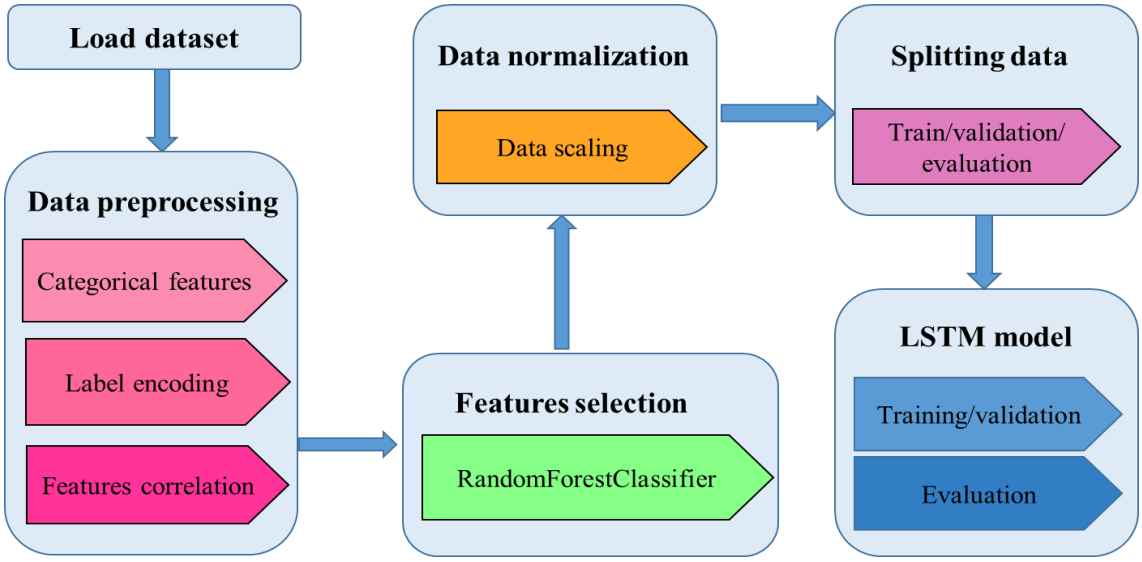

Figure 3. 2 : pré-processing et nettoyage des données.

Etape 1: Prétraitement des données (data preprocessing)

-Fonctionnalités catégorielles (categorical features): variables catégorielles sont des variables qui prennent des valeurs discrètes et non numériques ('proto', 'saddr', 'sport', 'daddr', 'dport'). Ils peuvent poser un défi lors de la modélisation, Et pour cela, nous les supprimons

-Codage d'étiquette (label encoding): La méthode fit_transform() de la classe LabelEncoder ajuste les catégories uniques dans chaque colonne et remplace les valeurs dans la colonne par des entiers. Nous l'avons utilisé pour transformer les valeurs textuelles des colonnes 'category' et 'subcategory' en des entiers.

-Corrélation des fonctionnalités (features correlation): Pour définir des caractéristiques importantes, supprimez les colonnes avec une corrélation supérieure à 0.8 (‘max’)

Étape 2: Sélection des fonctionnalités (features selection)

Nous utilisons la méthode de sélection de caractéristiques pour identifier les éléments les plus significatifs dans les données et ainsi limiter la quantité de données nécessaire pour la détection. Pour ce faire, nous avons recours à RandomForest Classifier, est avant tout un ensemble d'arbres de décision, où chaque arbre est une hiérarchie de questions if/else qui mènent à la prise de décision. Le seul inconvénient des arbres de décision est leur tendance à surajuster les données d'entraînement. Chaque arbre est différent de l'autre dans la forêt aléatoire. La théorie derrière les forêts aléatoires est que n'importe quel arbre prédira assez bien mais surajustera probablement certaines des données. Comme les autres arbres, tous fonctionnent bien et se surajustent de diverses manières, minimisant la sur-forme physique au moyen d'une moyenne des résultats. Cette baisse du surajustement préserve la capacité prédictive des arbres.

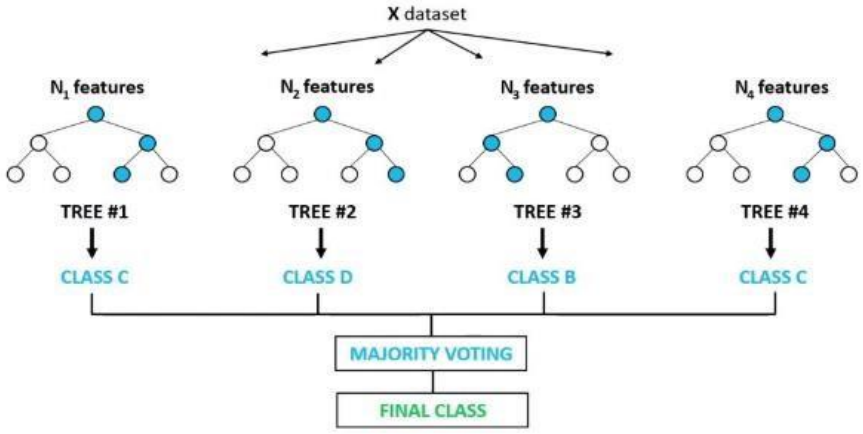

Figure 3. 3 : Random forest Classifier.

Étape 3: Normalisation des données (data normalization)

Dans cette étape, nous procédons à la normalisation des données en les mettant à l'échelle dans une plage de valeurs allant de 0 à 1. Pour ce faire, nous avons divisé l'ensemble de données en un ensemble d'entraînement et un ensemble de test à l'aide d'une validation croisée afin d'éviter un surajustement lors des étapes d'entraînement. Cette méthode nous permet de garantir que le modèle résultant est fiable et généralisable à de nouvelles données.

Étape 4: Division des données (splitting data)

Nous avons effectué une division aléatoire des données d'entrée X et y en deux ensembles distincts: un ensemble d'apprentissage (train) et un ensemble de test (test), en utilisant la fonction train_test_split() de la bibliothèque scikit-learn (sklearn), 20% des données sont réservées pour l'ensemble de test, tandis que les 80% restants sont utilisés pour l'ensemble d'apprentissage.

7.2. Modèle LSTM (Long Short-Term Memory)

Ce modèle contient une mémoire à court terme capable de durer assez longtemps pour qu'on la qualifie de mémoire longue à court terme. A l'aide des certaines bibliothèques nécessaires comme tensorflow et keras, nous avons créé notre modèle pour nos cas d'études avec les paramètres que nous avons cité dans le tableau 3.3.

7.2.1. Classification binaire

La première couche ajoutée au modèle est une couche LSTM. Cette couche est utilisée pour capturer les relations à long terme dans les données séquentielles. Elle comprend 7 neurones et prend en entrée des séquences de taille (7, 1).

Ensuite, une couche entièrement connectée est ajoutée avec 64 neurones. Cette couche permet d'apprendre des relations plus complexes entre les caractéristiques extraites par la couche LSTM.

La dernière couche utilise une fonction d'activation sigmoid pour générer une sortie binaire, indiquant une classification binaire. Cela suggère que le modèle est utilisé pour une tâche de classification binaire.

7.2.2. Classification multi classe

Dans cette classification, nous avons utilisé la même architecture de modèle, sauf la dernière couche entièrement connectée est ajoutée avec 4 neurones. Cette couche utilise une fonction d'activation softmax pour obtenir des probabilités de sortie, car le modèle semble être utilisé pour une tâche de classification avec 4 classes.

	Deep Learning		Federated Deep Learning	
Classification	Binary	Multi-class	Binary	Multi-class
Input	7	7	7	7
Output	1	4	1	4
Fonction d'activation	Sigmoid	softmax	Sigmoid	softmax
Optimisseur	Adam	Adam	Adam	Adam
Fonction de perte	binary crossentropy	categorical cross-entropy	binary crossentropy	categorical cross-entropy
Overffiting	K-fold Cross Validation Early stopping			

Tableau 3. 3: Paramètres du modèle LSTM utilisé.

Pour compiler le modèle nous avons utilisé plusieurs paramètres, ils sont présentés dans le tableau suivant:

	fonction de perte	optimisateur	métrique
Binaire	BinaryCrossentropy	Adam	accuracy
Multi-classe	Categorical_Crossentropy	Adam	accuracy

Tableau 3. 4: paramètre de compilation du modèle.

8. Scénarios

Pour l'identification des attaques dans le fog computing, nous avons utilisé les IDS basé sur la machine learning. Pour effectuer cette identification, nous avons utilisé le deep learning et le federated learning.
D'abord, nous avons commencé par la compilation de notre modèle qui se déroule de la manière indiquée dans le tableau 3.4.

Pour la classification binaire, nous avons utilisé fonction de perte BinaryCrossentropy, optimisateur adam et accuracy comme métrique. Par rapport à la multi-classe il s'agit de la fonction de perte Categorical_Crossentropy,optimisateur adam et métrique accuracy.

8.1. Scénario 1 : Deep learning

Dans cette section, nous avons appliqué les principes du deep learning, qui ont été précisément détaillés dans le chapitre 2.

Lorsque nous sommes passés à l'étape de l'entraînement, nous avons utilisé 20 epochs d'entraînement pour notre modèle dans la classification binaire et même multi-classe, nous avons utilisé l'algorithme de "Stratified K-fold Cross Validation" afin de prévenir le surajustement (overfitting).

8.2. Scénario 2 : Federated deep learning

Tout d'abord, nous avons mis en place une structure d'apprentissage fédéré, où les données et les modèles sont distribués parmi plusieurs clients. Les données d'entrée nous les avons divisées en 10 parties, tout comme les étiquettes correspondantes. Chaque client dispose également d'un modèle local qui a été initialisé avec la même configuration que le modèle initial.

Ensuite, nous avons procédé à une itération sur 5 époques le modèle global. À chaque époque, les poids globaux (global_weights) ont été initialisés pour chaque modèle local. Ensuite, chaque modèle local a été entraîné sur ses propres données pour 3 epochs. Les poids locaux (local_weights) ont ensuite été utilisés pour mettre à jour les poids globaux en utilisant une moyenne pondérée. Sans oublier que nous avons utilisé "Stratified K-fold Cross Validation"(il est bien défini dans le deuxième chapitre) .

9. Métrique d'évaluation (Binaire)

La “détection d'attaque” se concentre sur la détection des activités malveillantes dans le trafic IoT. Cela implique de classifier les flux de données en attaques ou non-attaques, en utilisant des algorithmes de détection d'anomalies ou de classification supervisée. L'objectif principal de la détection d'attaque est d'identifier les activités malveillantes et d'alerter les systèmes de sécurité pour prendre des mesures appropriées.

9.1. Matrice de confusion

L’utilisation de la matrice de confusion est importante pour évaluer de manière précise les performances des deux modèles utilisés.

- True Negatives(TN) : Le trafic normal est correctement identifié comme tel
- True Positives (TP) : le trafic d'attaque est identifié avec précision comme tel
- False Positives(FP) : le trafic normal est étiqueté à tort comme trafic malveillant
- False Negatives(FN): le trafic d'attaque est classé comme trafic normal lorsqu'il ne l'est

De plus, nous présentons la matrice de confusion afin de décrire les performances de classification de notre modèle. Cette matrice permet de récapituler les prédictions correctes et incorrectes obtenues en utilisant l'approche proposée, comme illustrée dans la figure suivant :

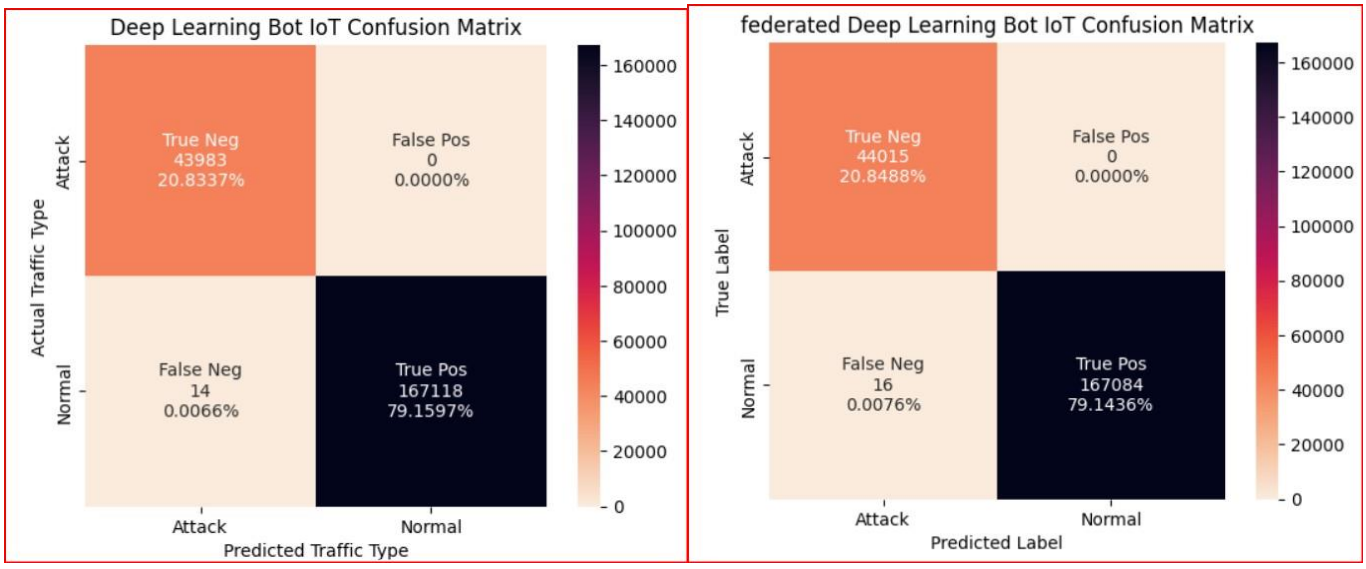

Figure 3. 4 : les matrices de confusion de deep learning et federated learning.

L'objectif est de fournir une évaluation quantitative des performances de notre modèle. Elle permet de minimiser le taux de faux positif et de faux négatif.

Par conséquent, notre cadre atteint un taux de faux positifs avec une valeur de 0% dans l'ensemble de nos données. Nous avons obtenu **0.007%** de faux négatifs dans federated learning et **0.006%** dans deep learning.

Par rapport au vrais Positifs (TP) nous avons obtenu **79.14%** dans le federated learning et **79.15%** dans le deep federated learning, et pour le vrai négatif (TF) **20.84%** dans le federated learning, et **20.83%** dans le deep learning.

Donc nous remarquons que la détection de faux négatifs dans le FL est légèrement meilleure que celle de DL.

9.2. La courbe ROC (Receiver Operating Characteristic)

La courbe ROC permet d'analyser et de comparer la performance des modèles de classification, en fournissant une visualisation et une mesure agrégée de leur capacité à discriminer entre les classes positives et négatives. La courbe ROC signifie l'ajustement entre le FPR (spécificité) et le TRP (sensibilité).

Notre modèle utilise à **99.99%** dans les deux scénarios, comme indiqué sur la figure 3.5:

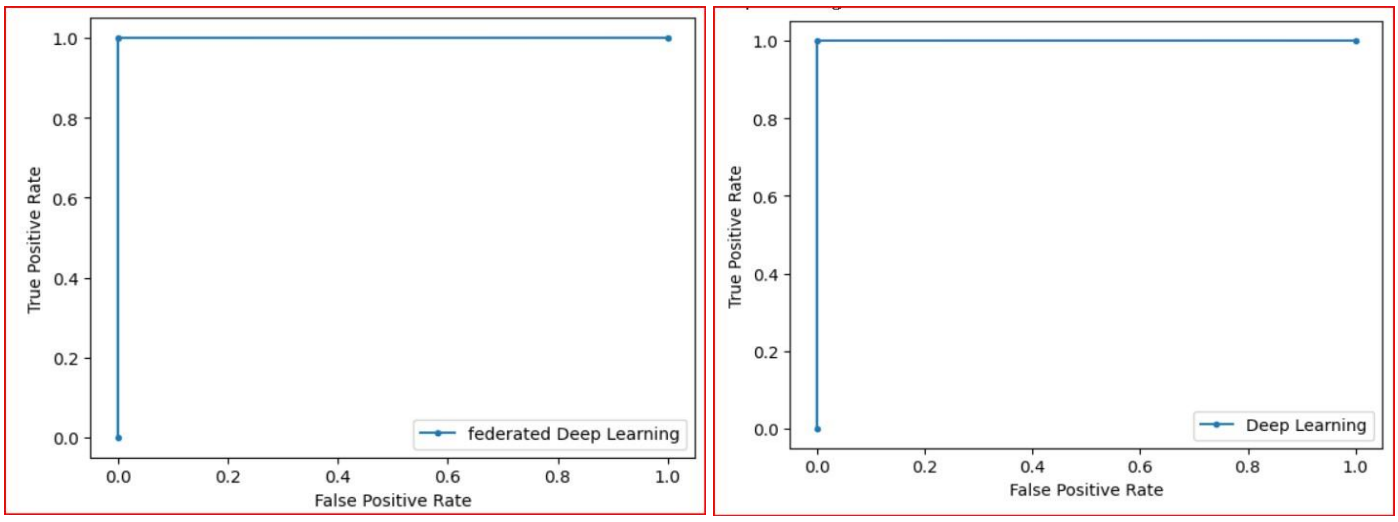

Figure 3. 5 : Les courbe ROC

9.3. Performance

Diverses mesures ont été utilisées pour évaluer le modèle proposé, notamment l'exactitude, la précision, le rappel et le score F. Cette évaluation comparative systématique avec d'autres approches pertinentes a permis de mieux appréhender les performances de notre modèle. Les mesures évoquées sont décrites ci-dessous:

9.3.1 Précision

Elle mesure la proportion d'observations positives prédites correctement parmi toutes les observations prédites positives.

Precision = tp /(tp + fp)

9.3.2. Rappel (recall)

Elle mesure la proportion d'observations positives prédites correctement parmi toutes les observations réelles positives.

recall = tp - score = tp /(tp + fn)

9.3.3. Score F1 (F1-score)

Elle mesure la moyenne pondérée de la précision et du rappel. Il est souvent utilisé comme une mesure globale de performance.

F1-score = fp / (fp - tn)

9.3.4. exactitude (Accuracy)

Elle désigne le rapport entre le nombre d'instances correctement classées et le nombre total d'échantillons observés.

accuracy = (tp + tn) /(tp + tn + fn + fn)

9.3.5 Evaluation des résultats de performance

Dans le but de valider notre système, nous avons réalisé une comparaison entre les IDS basé sur le deep learning et les IDS basé sur le federated learning, comme présenté dans le tableau suivant :

Method	Accuray	Précision	Recall	F1-score
ids basé sur fl	100	100	100	100
ids basé sur dl	100	100	100	100

Tableau 3. 5 :Comparaison entre les IDS basé sur le DLet les IDS basé sur le FL.

Au cours de cette analyse comparative, nous évaluerons les performances de chaque méthode.

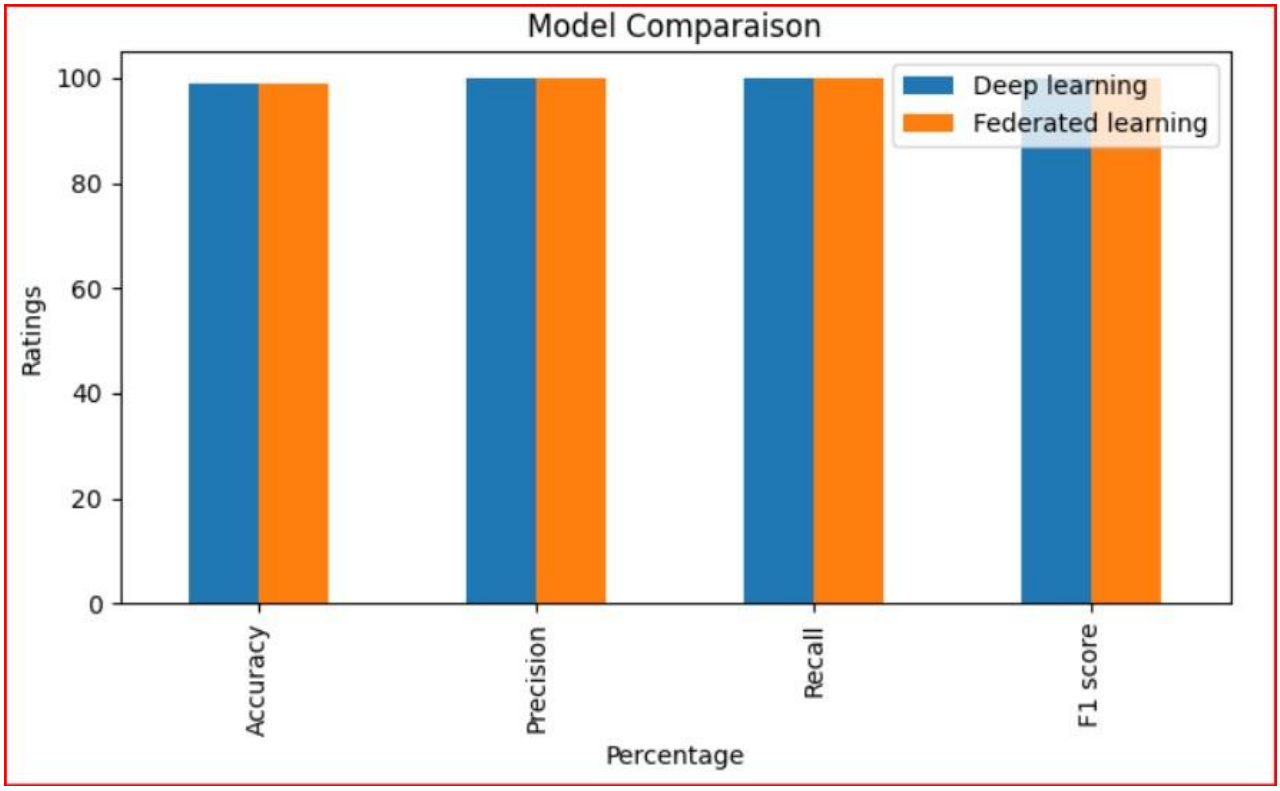

Figure 3. 6 : Evaluation des performances.

Comme le montre la figure 3.6 et le tableau 3.5, nous pouvons observer les performances de notre modèle. Quand on utilise le deep learning, les performances obtenues sont de **99.99** pour accuracy, **100%** pour recall et **100**% pour F1-score.

En revanche, avec le federated learning, les performances sont de **99.99** pour accuracy, **100**% pour recall et **100**% pour F1-score, et par rapport la précision Ils sont obtenu **100**%.

Nous pouvons observer que le federated learning et le deep learning ont donné de meilleurs résultats dans notre cas d'étude. Les scores obtenus sont élevés, indiquant une meilleure précision et une meilleure capacité de généralisation du modèle.

10. Métrique d'évaluation (multi-classe)

La "catégorisation" se réfère à la classification des attaques elles-mêmes en différentes catégories ou types. Dans notre cas, les attaques sont catégorisées en tant que déni de service (DoS), DDoS, normal, Reconnaissance.

10.1. Matrice de confusion

La matrice de confusion de deep learning et la matrice de confusion de federated deep learning pour la tâche de classification multiclasse sont illustrées à la Figure 3.7. Nous remarquons que les résultats de federated deep learning sont meilleurs que les résultats de deep learning.

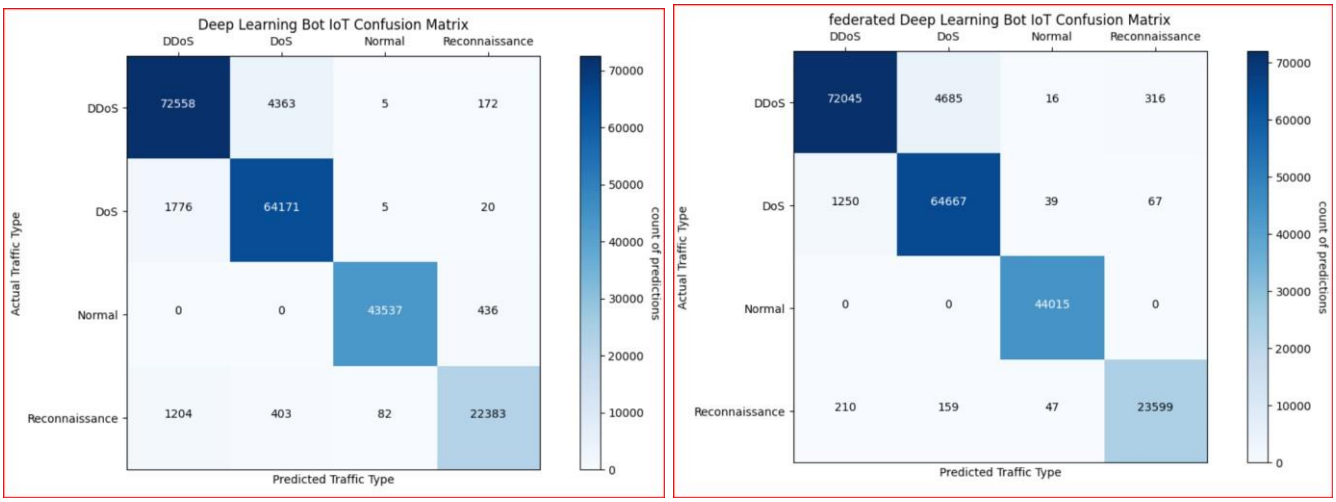

Figure 3. 7 : les matrices de confusion de deep et federated deep learning.

10.2. Courbe ROC (Receiver Operating Characteristic)

La courbe ROC est utilisée pour évaluer et comparer les performances des modèles dans une tâche de classification multiclasse. Comme indiqué précédemment, en offrant une représentation graphique et une mesure globale de leur aptitude à distinguer entre les classes positives et négatives. La courbe ROC représente la relation entre le taux de faux positifs (spécificité) et le taux de vrais positifs (sensibilité).

Notre modèle utilise à **96%** dans DDoS, **97%** dans DoS, **100%** dans Normal et **99%** dans Reconnaissance pour le federated learning.

Pour le deep learning notre modèle utilise à **96%** dans DDoS, **97%** dans DoS, **99%** dans Normal et **96%** dans Reconnaissance , comme indiqué sur la figure 3.8 et 3.9.

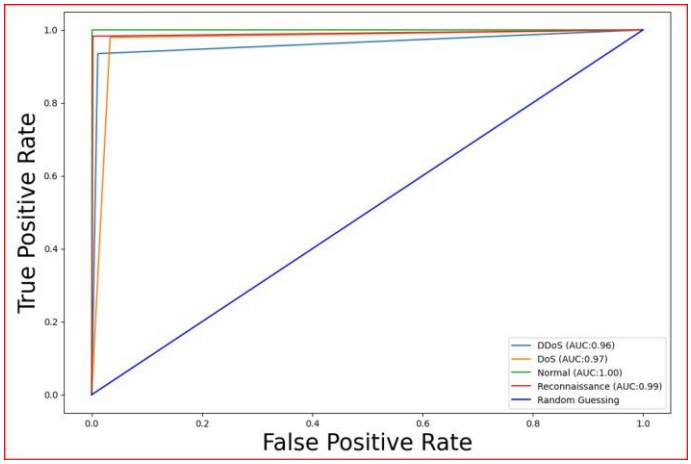

Figure 3. 8: courbes ROC de fdl.

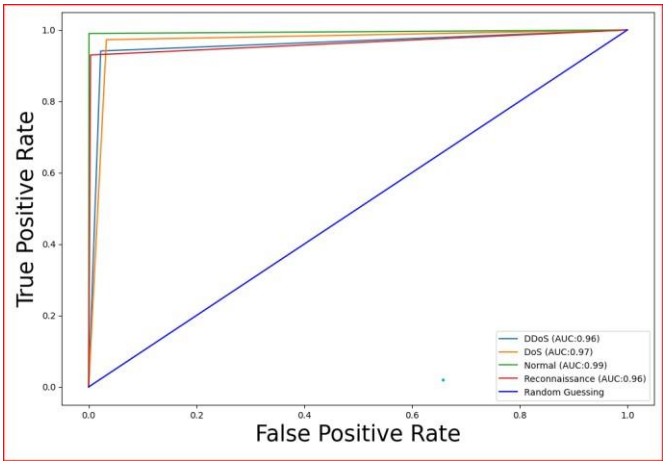

Figure 3. 9: courbes ROC de dl

10.3. Evaluation des résultats

Afin de valider notre modèle, nous avons effectué une comparaison entre les systèmes de détection d'intrusion (IDS) basés sur le deep learning et ceux basés sur le federated learning, comme illustré dans le tableau 3.6 ci-dessous :

Method	Catégorie	Accuray	Précision	Recall	F1-score
FDL	DDoS	97	98	93	96
	DoS		93	98	95
	Normal		100	100	100
	Reconnaissance		98	98	98
DL	DDoS	96	96	94	95
	DoS		93	97	95
	Normal		100	99	99
	Reconnaissance		97	93	95

Tableau 3. 6 : Comparaison entre les IDS basé sur le DL et les IDS basé sur le FDL.

Durant cette analyse comparative, nous procéderons à l'évaluation des performances de chaque méthode dans chaque catégorie.

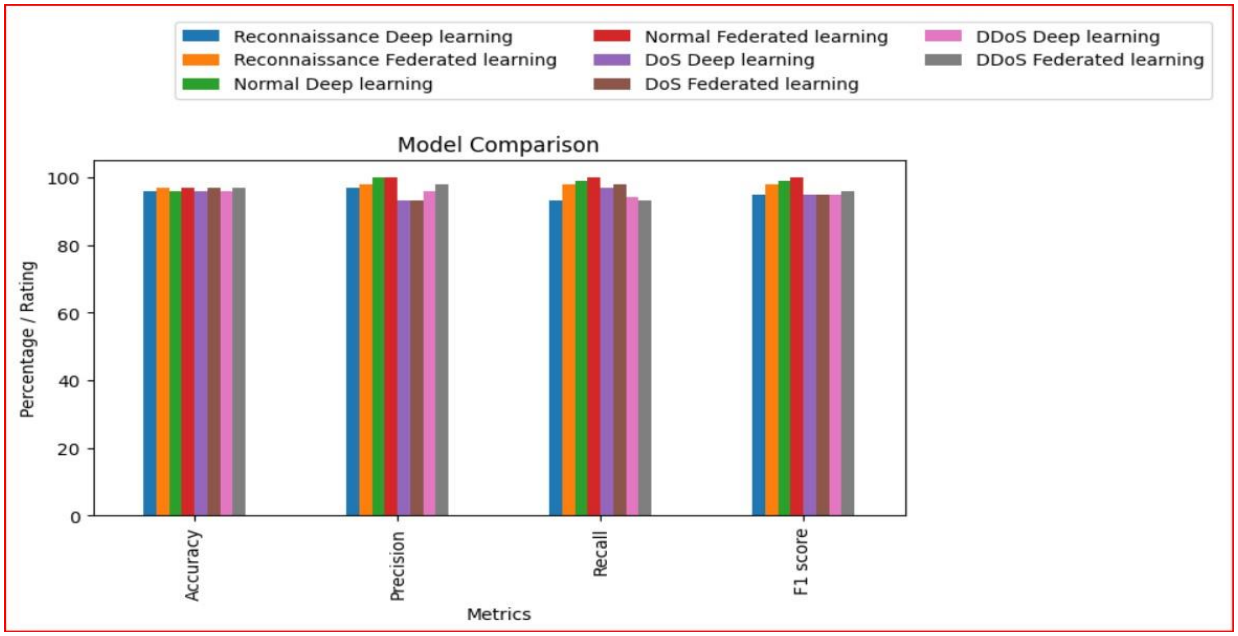

Figure 3. 10 : évaluation des performances.

Comme le montre les figures 3.10 et le tableau 3.6, nous pouvons observer les performances de notre modèle comme suite:

Pour l 'attaque DDos nous avons obtenu pour accuracy, precision, recall et f1-score pour federated learning les valeures respectivement de **97%, 98%, 93% et 96%.** Et pour deep learning les valeures de **96%**, **96%**, **94%** et **95%**.

Concernant l'attaque DoS nous avons marqué les valeurs **97%** ,**93%**, **98%** et **95%** pour federated learning, et pour le deep learning les valeurs **96%** ,**93%**, **97%** et **95%** successivement pour les performances accuracy ,precision ,recall et f1-score .

Dans le même ordre que les performances précédents , nous avons eu pour les attaques de **reconnaissance** des valeurs de **97%, 98% ,98% et 98%** dans le federated learning et **96%, 97% 93% et 95%** dans le deep learning , et pour **Normal** des valeurs de **97%, 100%, 100%** et **100%** dans le federated learning et **96%, 100%, 99%** et **99%** dans le deep learning.

Il est notable que, dans notre cas d'étude, le federated learning a généralement affiché des résultats supérieurs au deep learning. Les scores obtenus avec le federated learning sont globalement plus élevés, témoignant d'une précision accrue et d'une meilleure capacité de généralisation du modèle.

Conclusion

Dans ce chapitre, nous avons présenté la méthodologie de travail adopté pour la réalisation de notre projet. Les résultats obtenus ont démontré que l'utilisation de l'apprentissage fédéré dans un IDS donne de meilleurs réultats par rapport au deep learning.

Conclusion générale

Les systèmes IoT génèrent une quantité énorme de données chaque jour. La protection de la vie privée de ces données et dispositifs connexes est une nécessité. Les systèmes de détection d'intrusion (IDS) basés sur l'apprentissage automatique visent à répondre à cette nécessité. Mais ces cadres font toujours face à des problèmes pour répondre aux exigences de confidentialité et de sécurité en raison de leur nécessité de stocker et de communiquer des données au serveur centralisé. L'utilisation de solutions d'apprentissage fédérées IDS former modèle centralisé de haute qualité assurant la confidentialité des utilisateurs.

Ce projet a examiné l'utilisation de l'apprentissage fédéré dans le contexte de la détection des intrusions (IDS) dans le fog computing. Cette intégration offre une solution prometteuse pour sécuriser les systèmes IoT. Elle permet une détection efficace des intrusions tout en préservant la confidentialité des données, réduisant la charge de communication et améliorant l'adaptabilité du système aux environnements IoT dynamiques. Ces avancées ouvrent de nouvelles perspectives pour la sécurisation des systèmes IoT dans le contexte du fog computing et favorisent le déploiement de solutions robustes et efficaces de détection des intrusions.

Comme perspectives, il serait intéressant d'étudier la problématique de collaboration et de partage des connaissances : Les modèles d'IDS peuvent être améliorés de manière collaborative en intégrant les meilleures pratiques et les leçons apprises de différents nœuds de calcul, ce qui conduit à une amélioration continue de la détection des intrusions.

Références

[1]Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

[2]Singh, S., Jeong, Y. S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, *75*, 200-222.

[3]Saadi, M., Sayeh, M., & Khedim Allah, A. (2012). *Cloud Computing Application Aux Systèmes Mobiles Et Pair A Pair* (Doctoral dissertation, Université abderrahmane mira béjaia).

[4]Alzahrani, H. (2016). A brief survey of cloud computing. *Global Journal of Computer Science and Technology*, *16*(B3), 11-15.

[5]Apostu, A., Puican, F., Ularu, G., Suciu, G., & Todoran, G. (2013). Study on advantages and disadvantages of Cloud Computing–the advantages of Telemetry Applications in the Cloud. *Recent advances in applied computer science and digital services*, *2103*.

[6] https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html [Consulté le:09-avril2023]

[7]Iorga, B. M. G. M. Feldman. Fog computing conceptual model recommendations of the national institute of standards and technology. *NIST Special Publication*, 500-325.

[8]Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16).

[9]Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of network and computer applications*, *98*, 27-42.

[10]redha BOUAKOUK, M., Abdelli, A., & Mokdad, L. (2020, July). Survey on the Cloud-IoT paradigms: Taxonomy and architectures. In *2020 IEEE symposium on computers and communications (ISCC)* (pp. 1-6). IEEE.

[11]Prakash, P., Darshaun, K. G., Yaazhlene, P., Ganesh, M. V., & Vasudha, B. (2017). Fog computing: issues, challenges and future directions. *International Journal of Electrical and Computer Engineering*, *7*(6), 3669.

[12]Azizi, S., Shojafar, M., Abawajy, J., & Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. *Journal of network and computer applications*, *201*, 103333.

[13]Wang, P., Liu, S., Ye, F., & Chen, X. (2018). A fog-based architecture and programming model for IoT applications in the smart grid. *arXiv preprint arXiv:1804.01239*.

[14]Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015, November). Fog computing: Platform and applications. In *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)* (pp. 73-78). IEEE.

[15]Zaman, S., & Karray, F. (2009, August). Lightweight IDS based on features selection and IDS classification scheme. In *2009 international conference on computational science and engineering* (Vol. 3, pp. 365-370). IEEE.

[16]Whitter-Jones, J. (2018, April). Security review on the Internet of Things. In *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)* (pp. 163-168). IEEE.

[17]Mukherjee, M., Ferrag, M. A., Maglaras, L., Derhab, A., & Aazam, M. (2020). Security and privacy issues and solutions for fog. *Fog and fogonomics: challenges and practices of fog computing, communication, networking, strategy, and economics*, 353-374.

[18]Baïna, A. (2009). *Controle d'accès pour les grandes infrastructures critiques. Application au réseau d'énergie électrique* (Doctoral dissertation, INSA de Toulouse).

[19]Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*, 680-698.

[20]Jaafar, F. *et al.* (2023) 'Repérez le piratage : Systèmes de détection d'intrusion pour réseaux avioniques en utilisant l'apprentissage automatique', *Technologie et innovation*, 8(1).

[21]Khan, S., Parkinson, S., & Qin, Y. (2017). Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, *6*(1), 1-22.

[22] Hanifa, B. S. B. (2019). *Proposition d'un modèle de sécurité pour la protection de données personnelles dans les systèmes basés sur l'internet des objets* (Doctoral dissertation, Paris, CNAM).

[23]Saidi, A., Nouali, O., & Amira, A. (2022). SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and Fog computing. *Cluster Computing*, *25*(1), 167-185.

[24]Kamoun-Abid, F., Rekik, M., Meddeb-Makhlouf, A., & Zarai, F. (2021). Secure architecture for Cloud/Fog computing based on firewalls and controllers. *Procedia Computer Science*, *192*, 822-833.

[25] Bhatt, C., & Bhensdadia, C. K. (2017). Fog computing: applications, concepts, and issues. *International Journal of Grid and High Performance Computing (IJGHPC)*, *9*(4), 105-113..

[26] Birhanie, H. (2019). *Resource Allocation in Vehicular Fog Computing for an Optimal Use of EVs Electric Vehicles Energy* (Doctoral dissertation, Université Bourgogne Franche-Comté).

[27] Tsukerman, E. (2020). What Is an Intrusion Detection System (IDS). Designing a Machine Learning Intrusion Detection System.

[28]Wanda, P., & Jie, H. J. (2020). A survey of intrusion detection system. *International Journal of Informatics and Computation*, *1*(1), 1-10.

[29]Walling, S., & Lodh, S. (2022, October). A Survey on Intrusion Detection Systems: Types, Datasets, Machine Learning methods for NIDS and Challenges. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.

[30]Borkar, A., Donode, A., & Kumari, A. (2017, November). A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS). In *2017 International conference on inventive computing and informatics (ICICI)* (pp. 949-953). IEEE.

[31] Saxena, A. K., Sinha, S., & Shukla, P. (2017, May). General study of intrusion detection system and survey of agent based intrusion detection system. In *2017 International conference on computing, communication and automation (ICCCA)* (pp. 471-421). IEEE.

[32]Liu, C., Gu, Z., & Wang, J. (2021). A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning. *Ieee Access*, *9*, 75729-75740.

[33]Mathivet, V. (2017). *L'intelligence artificielle pour les développeurs: concepts et implémentations en C#*. Éditions ENI..

[34]https://www.oracle.com/fr/cloud/narrow-intelligence-artificielle/ [Consulté le:09-Mai-2023].

[35]https://www.oracle.com/fr/artificial-intelligence/deep-learning-machine-learning-intelligenceartificielle.html.[Consulté le:09-Mai-2023]

[36] https://www.bial-r.com/2019/05/22/comprendre-le-machine-learning-et-le-deep-learning/ [Consulté le:09Mai-2023]

[37]Bouafia, N. (2020). *Classification efficace des vêtements de mode basée sur les approches: apprentissage automatique ML et apprentissage profond DL* (Doctoral dissertation, FACULTE MATHEMATIQUES ET INFORMATIQUE-DEPARTEMENT INFORMATIQUE-OPTION: Informatique Décisionnelle Et Optimisation).

[38]Pajankar, A., & Joshi, A. (2022). Introduction to Machine Learning with Scikit-learn. In *Hands-on Machine Learning with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch* (pp. 65-77). Berkeley, CA: Apress.

[39]Lannes, L. M. P. (2022). *Unsupervised Learning Applied to the Segmentation of Users of Online Gambling Platforms in Portugal-The effects of the Covid-19 Pandemic on User Behavior and Segmentation* (Doctoral dissertation).

[40]Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015, June). Unsupervised learning of video representations using lstms. In *International conference on machine learning* (pp. 843-852). PMLR.

[41]Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. *Machine learning*, 3-23.

[42]Ceccon, F., Jalving, J., Haddad, J., Thebelt, A., Tsay, C., Laird, C. D., & Misener, R. (2022). OMLT: Optimization & machine learning toolkit. *The Journal of Machine Learning Research*, *23*(1), 15829-15836.

[43]rédac, T. (2022) *Machine learning : Définition, Fonctionnement, Utilisations, Formation Data Science | DataScientest.com*. Disponible au: https://datascientest.com/machine-learning-tout-savoir (Accessed: 22 June 2023).

[44]Hussein, S. A., Mahmood, A. A., & Oraby, E. O. (2021). Network Intrusion Detection System Using Ensemble Learning Approaches. *Technology*, *18*, 962-974.

[45]Illy, P., Kaddoum, G., Moreira, C. M., Kaur, K., & Garg, S. (2019, April). Securing fog-to-things environment using intrusion detection system based on ensemble learning. In *2019 IEEE wireless communications and networking conference (WCNC)* (pp. 1-7). IEEE.

[46]https://www.nvidia.com/en-us/glossary/data-science/deep-learning/ .[Consulté le:11-Mai-2023]

[47]https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagationmachine-learning-made-easy-e5285bc2be3a [Consulté le:20-Mai-2023]

[48]Touzet, C. (1992). *les réseaux de neurones artificiels, introduction au connexionnisme*. Ec2.

[49]Ullah, I., & Mahmoud, Q. H. (2022). Design and development of RNN anomaly detection model for IoT networks. *IEEE Access*, *10*, 62722-62750.
[50]Lee, S. W., Mohammadi, M., Rashidi, S., Rahmani, A. M., Masdari, M., & Hosseinzadeh, M. (2021). Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *Journal of Network and Computer Applications*, *187*, 103111.
[51]Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 4396.
[52]HAMOUDA, D. (2020). Un système de détection d'intrusion pour la cybersécurité.
[53]https://larevueia.fr/quest-ce-quun-reseau-lstm/ .[Consulté le:23-Mai-2023]
[54]https://developer.nvidia.com/discover/artificial-neural-network[Consulté le:25-Mai-2023]
[55]https://www.ummto.dz/dspace/bitstream/handle/ummto/12857/TayebMhamed.pdf?sequence=1 [Consulté le:25-Mai-2023]
[56]https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explicationsimple/#Les_differentes_fonctions_dactivation.[Consulté le:28-Mai-2023]
[57]Yadav, K. (2021). A Comprehensive Study on Optimization Strategies for Gradient Descent In Deep Learning. *arXiv preprint arXiv:2101.02397*.
[58]Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.
[59]https://medium.com/geekculture/loss-functions-and-optimizers-in-ml-modelsb125871ff0dc#:~:text=In%20machine%20learning%2C%20a%20loss,to%20minimize%20the%20loss%20 function[Consulté le:28-Mai-2023].
[60] https://datafranca.org/wiki/Optimisation_Adam [Consulté le:02-juin-2023]
[61]Bellahmer, H. (2020). *Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de propriétés immobilières* (Doctoral dissertation, Université Mouloud Mammeri).
[62]https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_565 [Consulté le:02-juin-2023].
[63]https://pub.towardsai.net/keras-earlystopping-callback-to-train-the-neural-networks-perfectly-2a3f865148f7 [Consulté le:02-juin-2023]
[64]Selamnia, A., Brik, B., Senouci, S. M., Boualouache, A., & Hossain, S. (2022, December). Edge Computingenabled Intrusion Detection for C-V2X Networks using Federated Learning. In *GLOBECOM 2022-2022 IEEE Global Communications Conference* (pp. 2080-2085). IEEE.
[65]Yang, L., Zhang, J., Chai, D., Wang, L., Guo, K., Chen, K., & Yang, Q. (2022, July). Practical and Secure Federated Recommendation with Personalized Mask. In *International Workshop on Trustworthy Federated Learning* (pp. 33-45). Cham: Springer International Publishing.
[66]Zhang, X., Zhao, X., Wu, Y., Zheng, H., & Li, Y. (2022). Federated Learning for Medical Image Classification: Advances, Challenges and Opportunities. *Challenges and Opportunities*.
[67]https://www.researchgate.net/figure/Federated-Learning-System-Horizontal-3_fig4_341083165 [Consulté le:06-juin-2023]
[68]https://www.researchgate.net/figure/Architecture-for-a-vertical-federated-learning-system_fig3_331086697 [Consulté le:09-juin-2023]
[69]Chen, Y., Qin, X., Wang, J., Yu, C., & Gao, W. (2020). Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, *35*(4), 83-93.
[70]https://www.researchgate.net/figure/Federated-Transfer-Learning-FTL-architecture_fig6_354427822 [Consulté le:02-juin-2023]
[71]Thapa, C., Chamikara, M. A. P., & Camtepe, S. A. (2021). Advancements of federated learning towards privacy preservation: from federated learning to split learning. *Federated Learning Systems: Towards NextGeneration AI*, 79-109.
[72]Elbir, A. M., Coleri, S., & Mishra, K. V. (2021, August). Hybrid federated and centralized learning. In *2021 29th European Signal Processing Conference (EUSIPCO)* (pp. 1541-1545). IEEE.
[73]https://www.researchgate.net/figure/Federated-learning-based-IDS-FLIDS_fig2_336568108 [Consulté le:02-juin-2023]

[74]Agrawal, S., Sarkar, S., Aouedi, O., Yenduri, G., Piamrat, K., Alazab, M., ... & Gadekallu, T. R. (2022). Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*.

[75]https://research.unsw.edu.au/projects/bot-iot-dataset [Consulté le:16-juin-2023]

[76]Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, *100*, 779-796.