République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et Recherche Scientifique
Université Saad Dahlab de Blida 1

Faculté des sciences
Département d'Informatique
En vue d'obtenir le diplôme de master
Domaine : Mathématique et informatique
Filière : Informatique
Option : Ingénieure de logiciel

Sélection des clients pour l'apprentissage fédéré dans l'Internet des objets

Présenté par :
- BENHASSINE Zineb
- BENHAMOU Kenza

Encadreur :
-M. AIECH Mohamed

Promoteur:
- Pr. ABED Hafida

Soutenu le : 06/07/2022

Devant le jury :

-M. BALA	Président
-Mme. DJEDAR	Examinateur

Année universitaire : 2021/2022

Remerciements

Tout d'abord, nous remercions Allah de nous avoir aidé et donné la force et la volonté de réaliser ce travail.

Ensuite, nous tenons à exprimer nos plus vifs remerciements et gratitude à notre promoteur Madame ABED Hafida pour son encadrement continu, pour les remarques constructives qu'elle nous a fournies ainsi que pour ses précieux conseils durant toute la période de notre travail. On la remercie également pour la confiance qu'elle nous a accordée et pour la grande liberté d'idées et de travail qu'elle nous a donnée. Nous n'oublierons pas aussi de la remercier pour ses qualités humaines, son hospitalité et son soutien qui ont permis de bien mener à bien ce travail.

Nous tenons à remercier les membres du jury d'avoir bien voulu participer à l'évaluation de ce travail.

Quelques personnes ont contribué à la réalisation de ce travail et méritent des remerciements.

Nous sommes également redevables à Mr Aiche Mohamed, ainsi qu'à Monsieur Bersali Mahmoud, tous deux doctorants au niveau du département d'informatique pour leur aide et conseils.

Enfin, nous tenons à remercier nos familles pour leur encouragement, leur aide et leur grande patience avec nous.

Dédicaces

A mon cher papa Nourddinne Benhamou, à ma chère maman Houria Benhamou, à ma chère sœur Sabrine, à mes chers frères Amine et Mahrez.

A la mémoire de mes grandes mères paternelle et maternelle : Hadjila et Ouardia qui ont été toujours dans mon esprit et dans mon cœur, je vous dédie aujourd'hui ma réussite. Que dieu, le miséricordieux, vous accueille dans son éternel paradis.

Je n'oublie pas aussi mon binôme Benhassine Zineb pour son travail, sa patience ainsi qu'à sa famille.

Je ne pourrais jamais exprimer le respect que j'ai pour vous. Vos prières, vos encouragements et votre soutien m'ont toujours été d'un grand secours. Puisse dieu, le tout puissant, vous préserver du mal, vous combler de santé et de bonheur et vous procurer une longue vie.

Kenza.

Dédicaces

Je dédie cet ouvrage

A ma chère maman Ahmed Allel Naima et mon cher papa Benhassine Mrizak qui m'a soutenu et encouragé durant ces années d'études. Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.

A ma sœur Wissam, mes frères, mes grands-mères, mes tantes et ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A ma famille, mes proches, et surtout ma tantes Ahmed Allel Salima et ma cousine Lamri Chahinez et sa tante Lamri Assia qui me donnent de l'aide, de l'amour et de la vivacité.

Sans oublier mon binôme Benhamou Kenza pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que Jaime.

Zineb.

Table des matières

Liste des figures

ملخص

التعلم الموحد هو نموذج ذكاء اصطناعي يسمح لعدد كبير من العملاء (كائنات متصلة) بموارد محدودة للتعاون من أجل تدريب نموذج دون مشاركة البيانات. درست العديد من الأعمال التعلم الفيدرالي مع مراعاة عدم تجانس البيانات وحدود الاتصال والحساب والمشاركة الجزئية للعملاء. ومع ذلك، فهم يفترضون مشاركة العملاء غير المتحيزة، حيث يتم اختيار العملاء بشكل عشوائي أو بما يتناسب مع حجم بياناتهم. قمنا في عملنا بمراجعة أحدث الأعمال في هذا المجال. أخذنا في الاعتبار مزايا وعيوب هذا العمل ثم اقترحنا نهجًا جديدًا لاختيار العملاء للتعلم الموحد.

في نهجنا، استخدمنا الخوارزمية التي تعطينا مجموعتين كناتج: الأول يضم العملاء بأفضل الوسائل والأداء. والثاني يشمل العملاء "الأسوأ" أو ذوي المهارات الضعيفة. لاختيار النقط الوسطى، نحسب قيم النقطة الوسطى الأمثل باستخدام طريقة التحسين التوافقي البسيط المزدوج.

الكلمات المفتاحية: التعلم الموحد، إنترنت الأشياء، التعلم العميق.

Résumé :

L'apprentissage fédéré est un paradigme de l'intelligence artificielle qui permet à un grand nombre de clients (objets connectés) ayant des ressources limitées de coopérer afin d'entrainer un modèle de deep learning sans partager les données. Plusieurs travaux ont étudié l'apprentissage fédéré en tenant compte de capacite CPU, temps d'apprentissage, taille de la RAM et enfin la fréquence. Cependant, ils supposent une participation impartiale des clients, qui sont sélectionnés au hasard ou en proportion de la taille de leurs données.

Dans notre cas d'étude, la domotique, une sélection des clients est primordiale. En effet, dans le cas de la détection des fuites de gaz, si le modèle global ne détecte pas les fuites rapidement, cela engendrera des pertes de vie humaine.

Dans notre approche, nous avons utilisé l'algorithme du K-Means qui nous donne en sortie deux clusters : le premier comportant les clients ayant les meilleurs moyens et performances. Le deuxième comportant les « pires » clients ou ceux ayant de mauvaises compétences. Pour le choix des centroïdes, nous calculons les valeurs du centroïde optimal en utilisant une méthode d'optimisation combinatoire le *dual de simplexe*.

Mots Clés : Apprentissage fédéré, Internet des objets, Apprentissage automatique.

Abstract:

Federated learning is an artificial intelligence paradigm that allows a large number of clients (connected objects) with limited resources to cooperate in order to train a deep learning model without sharing data. Several works have studied federated learning taking into account CPU capacity, learning time, RAM size and finally frequency. However, they assume unbiased participation from customers, who are selected randomly or in proportion to the size of their data.

In our case study, home automation, customer selection is essential. Indeed, in the case of the detection of gas leaks, if the global model does not detect the leaks quickly, this will cause loss of human life.

In our approach, we used the K-Means algorithm which gives us two clusters as an output: the first comprising the customers with the best means and performance. The second includes the "worst" customers or those with poor skills. For the choice of centroids, we calculate the values of the optimal centroid using a combinatorial optimization method the simplex dual.
Keywords: Federated Learning, Internet of Things, Deep Learning.

Chapitre I:
Introduction générale

1. Introduction :

Un monde rempli d'informations numériques sont aujourd'hui générées non seulement à partir des smartphones, mais aussi à partir d'appareils Internet des objets (IoT), en accédant aux données et en les partageant en déplacement. Ces énormes données sont utilisées pour former des modèles d'apprentissage automatique plus robustes et produire des applications plus intelligentes.

L'apprentissage fédéré (FL) est une nouvelle approche de l'apprentissage machine (automatique) qui permet aux clients (ex. appareils mobiles ou d'autres entités participantes ayant un dataset bien interessant) d'entrainer ou de former collaborativement un modèle sous l'orchestration d'un serveur central, tout en gardant les données d'entraînement décentralisées. Les données brutes de chaque client sont stockées localement et ne sont ni échangées ni transférées. Ce concept a été introduit en 2016 par McMahan et al. [15] et depuis beaucoup de travaux s'y intéressent.

L'Internet des objets (IoT) permet le développement d'une large gamme de produits, utiles dans la vie quotidienne dans des contextes tels que l'éducation, la santé, le commerce, le tourisme, l'agriculture, l'environnement, le transport et la domotique…etc. Dans ce mémoire nous sommes intéressés à la domotique, parce qu'elle s'agrandit et que le nombre d'appareils IoT de ce domaine devient plus en plus important, ceci nous aide beaucoup dans la partie étudiée dans cette recherche. la domotique offre un confort aux résidents grâce à une gamme de technologies qui permet l'automatisation, l'hébergement des équipements et une meilleure gestion de son environnement, elle assure également la sécurité de l'habitat et apporte le divertissement numérique.

2. Problématique :

Google a créé le Federated learning dans le but d'éliminer le problème de la confidentialité, c.-à-d., afin de préserver la vie privée des clients. La sélection des clients, particulièrement, a été le sujet de plusieurs recherches, et bien plus, diverses méthodes sont appliquées dans le but d'optimiser la sélection. En effet, les clients participants dans le FL ont une importance capitale dans ce

processus, selon leurs capacités, ils peuvent nuire à la construction du modèle global, comme ils peuvent être bénéfiques. Dans le domaine de la domotique, il existe plusieurs types de capteurs IOT ex. : capteurs de détection des fuites de gaz, détection d'intrus...

Ce travail vise à explorer les différents travaux existants dans ce cadre, et d'en y inférer les propositions les plus pertinentes pour la proposition d'un nouveau modèle et la mise en place d'une approche de sélection des clients qui soit tout aussi efficace, rapide et robuste afin d'éviter les pertes humaines qui pourraient être engendrer par la lenteur du modèle global ou par son manque de précision dans le cas de la fuite de gaz.

Nous proposons l'utilisation des méthodes de modélisation et de paradigmes de programmation nouvelle pour simplifier, décrire et résoudre le problème de la sélection des clients.

3. Objectifs :

Ce travail a pour but de trouver :

- En premier lieu, les critères les plus pertinents dans une sélection de client dans le domaine de la domotique.
- Proposer une nouvelle méthode de sélection qui soit robuste, efficace à la l'augmentation du nombre de clients.
- Formulation et résolution du problème de la selection.

4. Plan du mémoire :

Ce mémoire est constitué de cinq chapitres :

Nous commençons par présenter la problématique et les objectifs de ce projet de recherche.

Ensuite, dans le 2ème chapitre, nous exposons quelques méthodes de sélection présentées dans la littérature. Ces méthodes présentent différents aspects de sélection selon différents critères : premièrement, nous avons la sélection aléatoire, suivie d'une autre en s'appuyant sur la durée de l'apprentissage global que prend un client, ensuite vient la sélection aux multicritères qui utilise la régression linéaire et enfin la sélection selon la précision du modèle.

Le 3ème chapitre decrit notre approche qui consiste à sélectionner les clients aux multicritères en utilisant l'algorithme d'apprentissage automatique non supervisé K-Means.

Dans le 4ème chapitre, nous exposons les résultats obtenus après avoir simulé et comparé trois méthodes de sélection : aléatoire, selon temps d'apprentissage et enfin notre proposition : la sélection des clients aux multicritères en utilisant l'algorithme d'apprentissage automatique non supervisé K-Means. Finalement, nous terminons notre travail par une conclusion.

Chapitre II:
Etat de l’art

1. Introduction :

L'intelligence artificielle est une discipline de l'informatique qui a pour but de créer des machines intelligentes, en "opposition" avec l'intelligence naturelle des êtres vivants. Elle englobe toutes les idées visant à permettre à une machine de pouvoir émuler les capacités cognitives de l'Homme et de les surpasser. Ce terme "d'intelligence artificielle" voit le jour en 1956 après les nombreux travaux débutés après la Seconde Guerre Mondiale et constitue l'un des plus récents champs d'études parmi les sciences et l'ingénierie. Ceci a permis à de nombreuses inventions au cours des siècles de voir le jour. Les machines calculent des hypothèses sur lesquelles elles peuvent réfléchir et agir [1].

Le terme a beaucoup évolué au fil du temps avec l'émergence de l'apprentissage automatique, l'apprentissage profond et puis l'apprentissage fédéré qui sont expliqués dans ce qui suit. la figure 1 explique clairement l'inclusion de l'apprentissage dans l'intelligence artificielle.

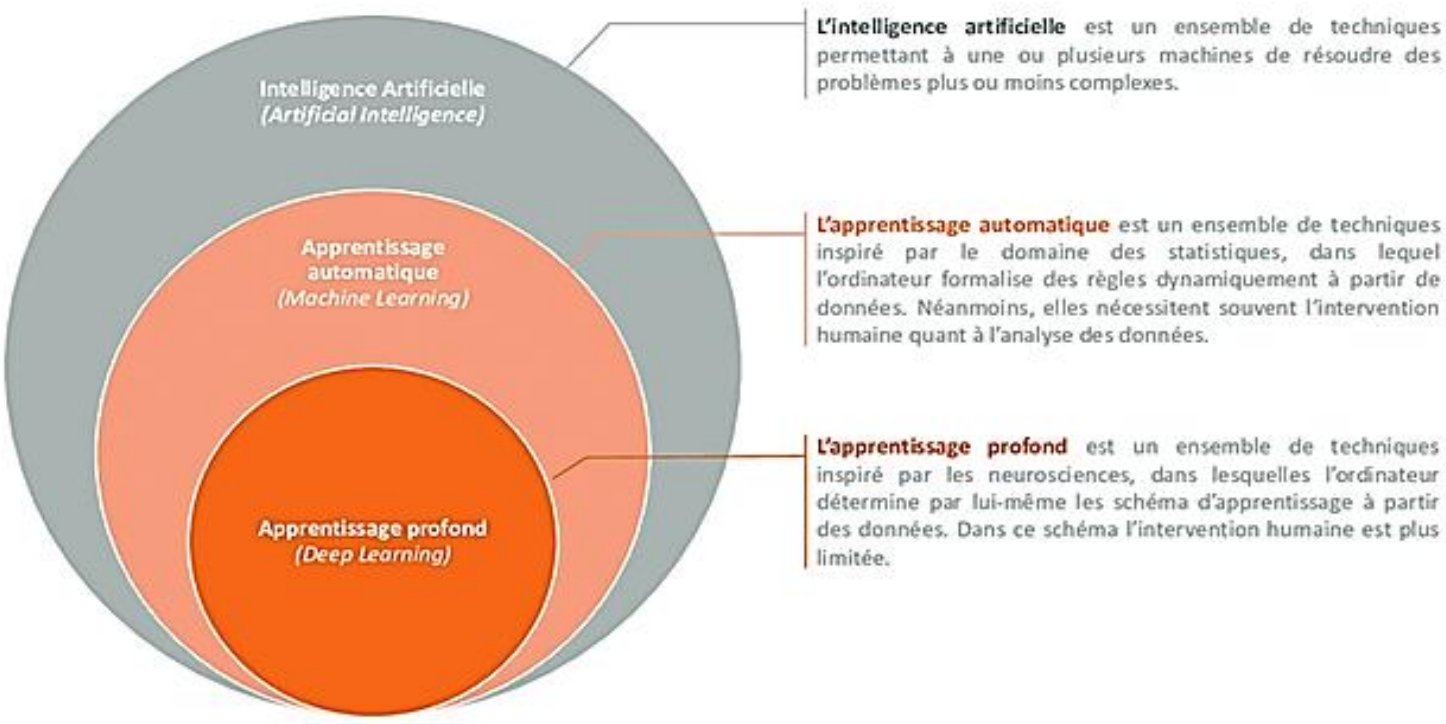

Figure 1: Schéma explicatif du domaine de l'IA[11].

Cette discipline est appliquée, par la suite, sur des robots ou des objets de différents domaines tel que le domaine de la santé, la sécurité, la domotique…

Ces objets sont nommés Internet des objets pour leur connexion à internet, plus précisément au cloud, et ces derniers vont être introduits plus en détails dans le prochain paragraphe.

2. Internet des objets :

1.1. *Définition des systèmes IoT :*

La définition de l'internet des objets (IoT) n'est pas explicite et ne connait pas une définition qui soit précise, cependant, plusieurs définitions descriptives et par utilité existent. Les auteurs dans [2] Ont fait une recherche taxonomique sur l'IoT et proposent la définition suivante :

un groupe d'infrastructures interconnectant des objets connectés, permettant leur gestion, exploration des données et accès aux données qu'ils génèrent où les objets connectés sont des capteurs ou des actionneurs effectuant une fonction spécifique et capables de communiquer avec d'autres équipements.

D'autres définitions existent, telle que celle de [3] :

qui le définit comme étant, une extension de la connectivité réseau et de la capacité informatique aux objets, dispositifs, capteurs et éléments qui ne sont pas généralement considérés comme des ordinateurs. Ces objets intelligents nécessitent une intervention humaine minimale pour générer, échanger et consommer des données. Ils offrent souvent une connectivité aux fonctions de collecte, d'analyse et de gestion de données à distance [3].

Le terme "Internet des objets" (internet of things) a été introduit en 1999 par Ashton, K. [4]. Il explique que les ordinateurs ont besoin de l'intervention de l'humain pour l'acquisition de l'information sur le monde réel et la réalisation des tâches. La société, l'économie et la survie sont basés sur des "objets" [4].

Les concepts associés à l'IoT incluent:

- M2M (Machine to Machine): Le concept de M2M (communication intermachines) est apparu dans les années 70 dénotant un ensemble de

machines connectées entre elles et pouvant échanger de l'information sans l'intervention humain, à l'aide des solutions de télécommunication PPP (liaison point à point) ou des moyens spécifiques aux fabricants. Elles peuvent faire du reporting (envoyer des rapports) vers un serveur M2M dédié au monitoring (surveillance). Contrairement à l'IoT, où les objets connectés sont hétérogènes et la connexion étant standardisée par un protocole (TCP/IP), les entités connectées doivent soit être homogènes (même protocole de communication : HDLC, PPP, SLIP, etc.), soit être construites par le même fabricant pour assurer la compatibilité et la compréhension des messages échangés. On en déduit que l'IoT est une forme évoluée, plus générale de la M2M [5].

- IoE (Internet of Everything): Le terme IoE (Internet of Everything) a été introduit par la société spécialisée dans le matériel réseau Cisco pour lancer un nouveau domaine du marketing. Il s'agit non seulement d'identifier et d'interconnecter les objets, y compris éventuellement les individus, mais aussi les connecter à l'internet. Ce domaine de marketing se matérialise par les compagnies de collecte d'information qui traitent les Big Data chargées d'en y retirer des bénéfices en la revendant à des entités de productions ou des services.

- WoT (Web of Things) : Un synonyme pour le terme IoE est le Web of Things (WoT) où les objets connectés utilisent une même base (le web) pour le stockage des informations, ce qui permet aux objets de communiquer les uns avec les autres [5].

1.2. *Architecture de l'internet des objets :*

Il s'agit d'un modèle qui organise l'Internet des objets en cinq couches différentes. La figure 2 montre la description de chacune d'elles [6] :
- Premièrement, la couche de perception regroupe tous les objets physiques. Elle collecte et rend les données numériques et les envoient à la couche supérieure via des canaux sécurisés.

- La seconde, c'est la couche réseau qui joue le rôle de connecter entre eux tous les équipements.
- La couche traitement repose essentiellement sur les technologies de «middleware » permettant de réunir les « hardwares » et « softwares » sur une

même plateforme.
- Enfin, la couche application offre la possibilité d'utiliser les informations traitées par la couche traitement pour les développer.

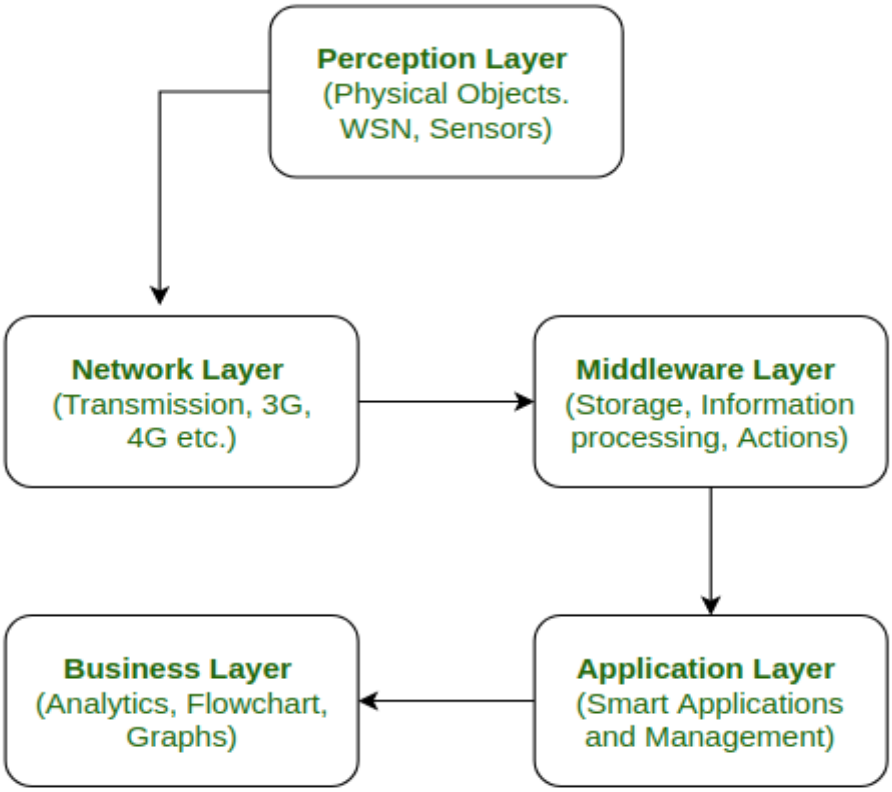

Figure 2: Schéma d'une architecture IoT[6].

1.3. *La domotique :*

L'Internet des objets (IoT) permet le développement d'une large gamme de produits, utiles dans la vie quotidienne dans des contextes tels que l'éducation, la santé, le commerce, le tourisme, l'agriculture, l'environnement, le transport et la domotique…etc. nous nous intéressons dans ce mémoire à la domotique, parceque le nombre d'appareils IoT de ce domaine devient plus en plus important, ceci nous aide beaucoup dans la recherche des clients pertinents. De plus, le but d'introduire l'IoT dans ce domaine est de rendre les appareils plus autonomes, ainsi notre méthode aide à améliorer l'apprentissage de ces derniers, tout en gardant la confidentialité de ces appareils. Par exemple, on prend bâtiment intelligent tel qu'un un centre commercial, les caméras de surveillance devront alerter les forces spéciales en cas d'intrusion dans les plus

brefs délais, entre temps ces derniers devront aussi garder confidentielles ce que l'utilisateur ne veut pas divulguer.

1.3.1. Définition de la domotique :

Le terme domotique vient du mot [domo (domus) + tick], qui signifie [Famille + Technologie ou informatique] Scientifiquement, La domotique ou smarthome désigne l'ensemble des technologies qui s'appliquent à l'habitat pour en améliorer le confort et faciliter les économies d'énergie. La domotique regroupe plusieurs types de technologie qui automatisent la maison ou l'entreprise, qui assurent également la sécurité de l'habitat et apporte le divertissement numérique. La domotique est un concept qui fait de plus en plus parler d'elle ces derniers temps, mais ce n'est pas un nouveau concept car il existe depuis les années 1980. Depuis, ces technologies n'ont cessé de se développer et de gagner en popularité [7].

1.3.2. Les maisons intelligentes :

Dans le domaine de la domotique, on trouve la maison intelligente, tel que illustré dans la figure 3, qui est une maison contrôlée et gérée via un ensemble de boutons et de télécommandes, ou des applications utilisant les protocoles internet via les réseaux locaux sans fil. Ce dernier permettra d'avoir une surveillance en continue de la maison [8].

Figure 3: Maison intelligente [8].

Cette invention représente l'avenir qui offre beaucoup plus de maîtrise aux personnes qui y habitent. La maison connectée apporte plus de sécurité, un confort indéniable et assure des économies d'énergie importantes [8].

Les maisons intelligentes (figure 3) ont la capacité d'augmenter le confort de l'habitant à travers : des interfaces naturelles pour piloter la lumière, la température ou les différents appareils électroniques. La gestion des ressources énergétiques est un autre enjeu des maisons intelligentes. Donc, c'est possible de mettre en veille les dispositifs de chauffage quand les habitants sont absents ou adapter automatiquement l'utilisation des ressources électriques en fonction des besoins des résidents afin de diminuer les gaspillages de ressources énergétiques [8].

Les bénéficiaires de ces innovations peuvent être des individus autonomes mais également des personnes fragiles ayant une capacité limitée de mouvement. Par exemple, les personnes âgées ayant une autonomie limitée pourraient profiter des applications des maisons intelligentes (figure 3) pour faciliter leur vie quotidienne ou rester en contact avec leurs proches. Actuellement, les changements démographiques provoqués par le vieillissement de la population et l'augmentation du nombre de personnes âgées vivant seules ont un impact social et économique important au sein de la société. À cet égard, l'usage de la technologie représente une grande opportunité pour les personnes âgées vivant seules. Les systèmes intelligents peuvent rappeler aux habitants lorsqu'ils doivent prendre leurs médicaments, faciliter leur mise en communication avec l'extérieur ou même alerter les proches ou le service d'urgence si la personne tombe par accident [8].

1.3.3. Les bâtiments intelligents :

Le concept du bâtiment intelligent est né aux USA et a évolué depuis les années 80, sans définition fixe ou standardisée. Il dispose dans la littérature de multiples définitions qui peuvent varier en fonction des cultures scientifiques mais l'objectif final reste l'amélioration du confort et de la productivité des occupants [9].

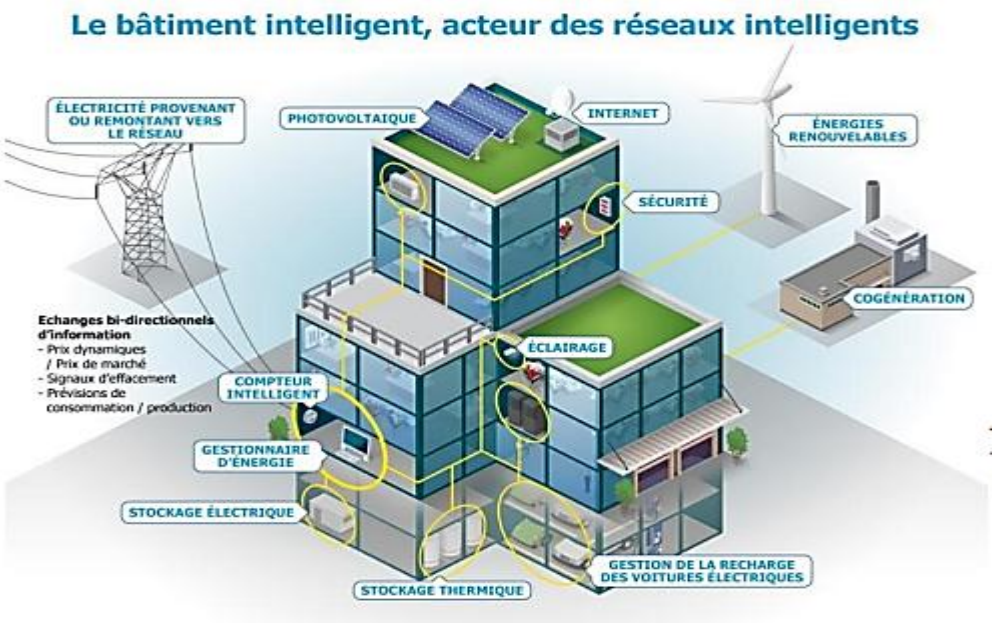

Figure 4: Le bâtiment intelligent au sein des réseaux intelligents[9].

Le concept de bâtiment intelligent (figure 4) relie les notions de domotique et de gestion de l'énergie au niveau domestique. Il est issu de la technologie de smart grids (ou réseaux intelligents). La gestion des réseaux électriques à l'aide des nouvelles technologies de l'information, mais adapté au réseau privé [10]

Les technologies présentent au sein du bâtiment intelligent (figure 4) permettent le contrôle de différents paramètres en vue d'optimiser la consommation d'énergie ainsi que le confort et la sécurité de l'usage [10].

Du point de vue de l'usage, l'ensemble des systèmes mis en place doit pouvoir permettre une utilisation alliant confort, sécurité et simplicité [10].

1.3.4. Les divers de la domotique :

1.3.4.1. Le confort :

En domotique, les appareils sont automatisés pour améliorer le confort de la résidence. Les capteurs de ces appareils peuvent être thermiques (figure5), ou contrôlés par des équipements vocaux tels que l'assistant google, etc [7].

A. Aspect thermique :

Dans une maison typique, il est difficile de s'adapter aux changements de température (chaude et froide) [7].

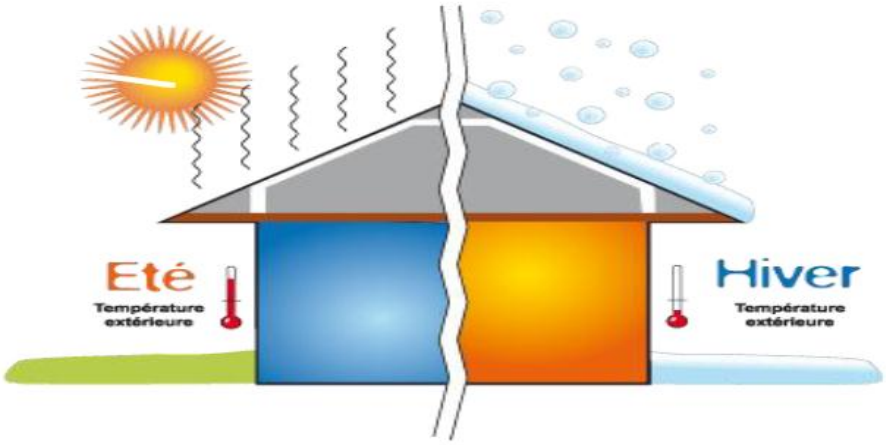

Figure 5: Le confort thermique dans une maison intelligente [7].

En revanche, dans une maison intelligente, nous pouvons automatiser le chauffage, la ventilation et la climatisation pour maintenir la maison à température ambiante à tout moment, grâce à l'utilisation du thermostat électrique (régulateur électronique). Avec la domotique, nous pouvons programmer, régler le chauffage et régler la température souhaitée, Indépendant pièce par pièce. La programmation peut être modifiée à tout moment, même d'une certaine distance. Par exemple, avant de quitter le travail, nous pouvons passer le mode chauffage de votre smartphone en mode confort [7].

B. Contrôle par la voix humaine :

Parmi les nouvelles technologies, la domotique peut contrôler les appareils par voix humaine. L'habitat doit répondre aux demandes des utilisateurs. Un ensemble d'actions peuvent-être automatisées (contrôle des lumières, baisse les stores, équipement multimédia, etc.) [7].

Pour y parvenir, les systèmes de contrôle à domicile doivent tenir compte du contexte dans lequel la commande est passée, ainsi que comprendre les habitudes et les préférences de l'utilisateur. [7] Par exemple, faire des recherches pendant des heures sur Smartphone sous les coussins du canapé pour activer la climatisation ou baisser les stores.

1.3.4.2. La sécurité :

La domotique nous permet de gérer notre maison, nos occupants et notre sécurité, autorisation d'accès par reconnaissance vocale, cartes magnétiques, codes numériques, talkies walkies, détecteurs de mouvement et appareils (protection incendie, protection contre les inondations, etc.) [7].

A. Des personnes :

Cet aspect concerne les personnes vulnérables (personnes âgées, handicapées, etc.) et les autres occupants. La figure 6 montre la domotique pour les services d'assistance à domicile [7] :

- Médaillon d'appel : placé autour du cou en cas d'un accident le pendentif permet à la personne âgée de contacter un opérateur de téléassistance.
- La télécommande : la personne pourra ouvrir la fenêtre, allumer la lumière sans sa place et sans déplacé.
- Un chemin lumineux automatique pour faciliter le déplacement pendant la nuit.
- Installer des capteurs qui vont progressivement intégrer le niveau d'activité de la personne âgée, en cas d'anomalie, des alertes seront déclenchés, comme on peut installer une caméra pour surveiller la personne au cas où une chute.

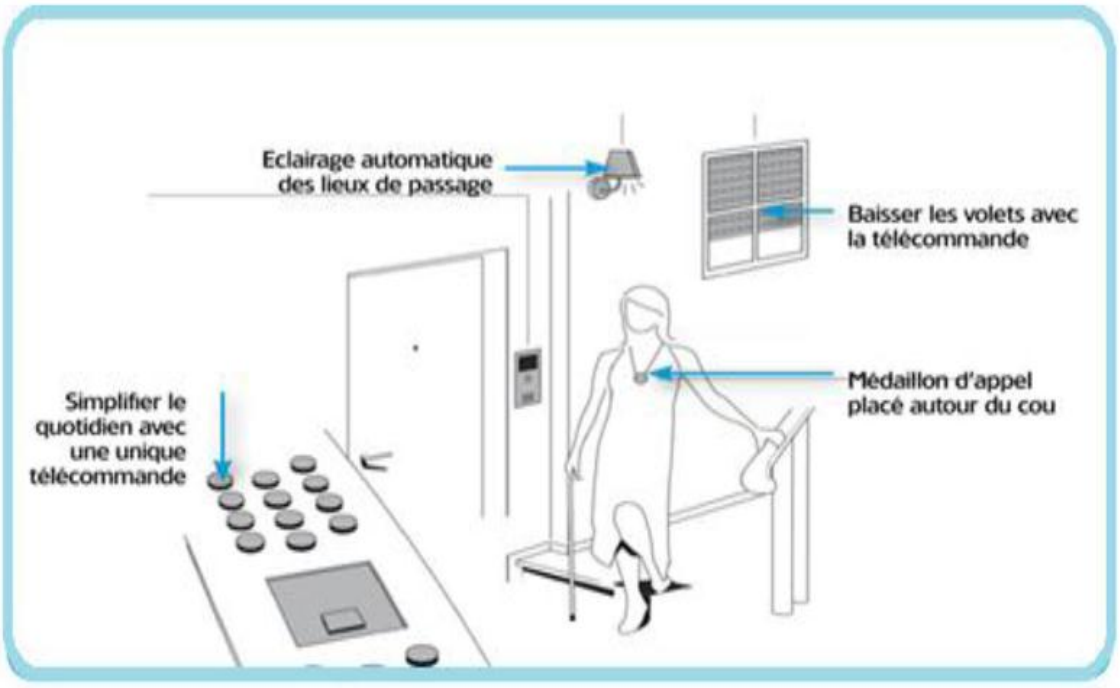

Figure 6: La domotique au service de l'assistance à l'autonomie à domicile[7].

B. Des biens

L'intrusion dans l'habitat peut être surveillée par [7]:

- Des capteurs placés sur les portes ou fenêtres.
- Des détecteurs de mouvement internes ou externes détecteront la présence de personnes.

Ces éléments pourront ensuite soit déclencher une alarme conventionnelle (sirène, feux clignotant, alerte vers un service d'intervention), soit envoyer un message ou une vidéo vers un Smartphone. Voici quelques exemples de sécurité [7] :

- La surveillance des équipements domestiques (fuite d'eau, de gaz, appareils restés branchés …). Ceci pourra se faire par une consultation à distance du statut de certains équipements ou par l'envoi automatique d'un message d'alerte.
- La détection d'incendie connectée au réseau domestique couvre aussi bien la protection des biens que des personnes. Parmi les solutions de la domotique (déclencher l'alarme, envoyer un message ou un SMS sur le téléphone, mettre en marche un éclairage pour guider les déplacements, lever les volets roulants pour une évacuation rapide…).

Comme pour les humains, les objets intelligents ont besoin d'apprendre, d'évoluer. Pour cela, des méthodes d'apprentissage ont été créés dans le but d'aider ces derniers à apprendre en se basant sur des connaissances apprises par des bases de données, et appliquée durant des tests.

Le prochain paragraphe va vous introduire les différents types d'apprentissage : automatique et profond avec tous leurs sous type : supervise et non supervisé.

3. L’apprentissage dans l’intelligence artificielle :

3.1. L’apprentissage automatique :

L’apprentissage automatique (machine) est la faculté d’apprendre de ses expériences passées et de s’adapter est une caractéristique essentielle des êtres humains. Elle est essentielle à l’être humain dans les premières étapes de la vie pour apprendre des choses aussi fondamentales que reconnaître une voix, un visage familier, apprendre à comprendre ce qui est dit, à marcher et à parler [11].

L’apprentissage automatique est une tentative de comprendre et reproduire cette faculté d’apprentissage dans des systèmes artificiels. Il s’agit très schématiquement, de concevoir des algorithmes capables, à partir d’un nombre important d’exemples (données , base de données, dataset…), d’apprendre afin de pouvoir appliquer ce qu’ils ont ainsi assimilé aux cas futurs [11].

Il existe deux sous-types d’apprentissage automatique. Ces derniers vont être présentés dans le prochain paragraphe.

3.1.1. Types d’apprentissage automatique :

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage employé [11]:

3.1.1.1. Apprentissage supervisé :

L'apprentissage supervisé a pour but d'établir des règles de comportement à partir d'une base de données contenant des exemples de cas déjà étiquetés. La base de données est en principe un ensemble de couples entrées / sorties {(X, Y)}. Le but est d'apprendre à prédire pour toute nouvelle entrée X, la sortie Y [11].

L’apprentissage supervisé utilise plusieurs méthodes tels que [11]:

- Boosting.
- Machine à vecteurs de support.
- Mélanges de lois.
- Les réseaux de neurones.
- Méthode des k plus proches voisins.
- Arbre de décision.
- Classification naïve bayésienne.

Ce type d'assimilation peut être appliqué dans plusieurs domaines, nous citons:

- Vision par ordinateur.
- Reconnaissance de formes.
- Reconnaissance de l'écriture manuscrite.
- Reconnaissance vocale.
- Traitement automatique de la langue.
- Bio-informatique.

3.1.1.2. Apprentissage non-supervisé :

Contrairement à l'apprentissage supervisé, le non supervisé traite le cas où on dispose seulement des entrées {X} sans avoir au préalable les sorties. L'apprentissage non supervisé ou le « clustering » vise à construire des groupes (clusters) d'objets similaires à partir d'un ensemble hétérogène d'objets [11].

On distingue plusieurs algorithmes de « clustering », comme [11]:

- **K-moyennes** (KMeans). KMeans est un algorithme de partitionnement des données en K nombre de groupes ou clusters. Chaque objet sera associé à un seul cluster. Le nombre K est fixé par l'utilisateur.
- **Fuzzy KMeans.** Il s'agit d'une variante du précédent algorithme proposant qu'un objet ne soit pas associé qu'à un seul groupe.
- **Espérance-Maximisation (EM**). Cet algorithme utilise des probabilités pour décrire qu'un objet appartient à un groupe. Le centre du groupe est ensuite recalculé par rapport à la moyenne des probabilités de chaque objet du groupe.
- **Regroupement hiérarchique.** Deux sous-algorithmes en découlent, à savoir d'une part le «Bottom up» qui a pour fonction d'agglomérer des groupes similaires, donc en réduire le nombre (les rendre plus lisibles) et d'en proposer un ordre hiérarchique, et d'autre part, le «Top down» qui fait le raisonnement inverse en divisant le premier groupe, récursivement, en sous-ensembles.
- **Neural networks (Réseaux de neurones).**

3.2. Apprentissage profond (Deep learning):

Le terme « apprentissage profond » a été introduit dans le domaine du l'apprentissage automatique en 1986, et dans les réseaux de neurones artificiels en 2000, dans le contexte des neurones à seuil booléen, l'apprentissage profond désigne une technique

d’apprentissage d’une machine, c’est une sous-branche de l’intelligence artificielle qui vise à construire automatiquement des connaissances à partir de grandes quantités d’information .Les caractéristiques essentielles du traitement ne seront plus identifiées par un traitement humain dans l’algorithme préalable, mais directement par l’algorithme d’apprentissage profond [12].

L’apprentissage en profondeur permet donc implicitement de répondre à des questions du type « que peut-on déduire de ces données ?» et décrire des caractéristiques parfois cachées ou des relations entre des données souvent impossibles à identifier pour l’homme [12].

L’apprentissage profond est un réseau neuronal avec un grand nombre de paramètres et de couches, l’exemple de base c’est le perceptron multicouche MLP « multi layer perceptron » (Voir figure 7) [12].

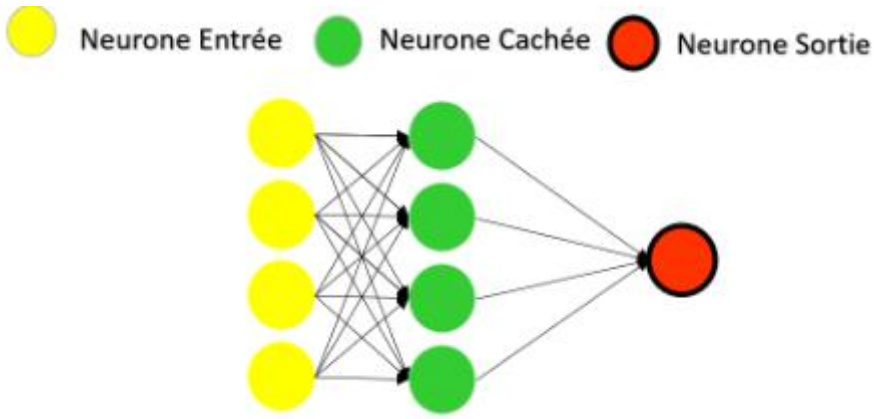

Figure 7: Un perceptron multicouche [12].

Perceptron a été inventé en 1958, le mot vient de verbe latin « Percipio » qui signifie en Anglais understand ; en Français comprendre, qui montre que Le robot ou l’appareil peut apprendre et comprendre le monde extérieur [12].

Un perceptron multicouche avec plusieurs couches cachées entre la couche d'entrée et la couche de sortie est un réseau de neurones profonds (DNN), le DNN est une fonction mathématique, qui mappe certains ensembles de valeurs d’entrée aux valeurs de sortie. La fonction est formée par la composition de nombreuses fonctions plus simples [12].

Certaines de ses caractéristiques [12]:

- Plus de neurones.
- Des moyens plus complexes de connecter les couches neurones dans les réseaux neuronaux.
- Puissance de calcul.
- Extraction automatique des fonctionnalités.

L'apprentissage profond s'applique dans divers domaines, tel que [12] :

- L'intelligence artificielle en général.
- La robotique.
- La santé et la bio-informatique.
- La sécurité.

L'application de ces méthodes d'apprentissage est purement centralisée, c-à-d : que les données, riches en quantité et en qualité, des appareils sont transmises au cloud afin de procéder à l'apprentissage. Cependant, il existe des données confidentielles que le propriétaire ne souhaite pas divulguer tels que les emails et les appels enregistrés…

[4] ont étudié une technique d'apprentissage qui permet aux utilisateurs de récolter collectivement les fruits de modèles partagés formés à partir de ces données riches, sans qu'il soit nécessaire de les stocker de manière centralisée. Ils nomment cette approche Federated Learning, puisque la tâche d'apprentissage est résolue par une fédération lâche d'appareils participants (qu'ils appellent clients) qui sont coordonnés par un serveur. Le prochain paragraphe va mettre la lumière sur la notion d'apprentissage fédérée.

4. Apprentissage Fédéré :

4.1. Définition :

L'apprentissage fédéré (FL) permet de former l'algorithme d'apprentissage automatique et de conserver les données au niveau des appareils. Cela signifie que le FL permet à chaque appareil de conserver ses propres données privées et locales. La méthode traditionnelle, tel que l'apprentissage centralisé par machine, n'incluait pas ces avantages et comportait un risque élevé pour la protection des données et le transfert de fichiers volumineux [13].

4.2. Origine [14] :

L'apprentissage fédéré d'abord a été pratiqué dans une architecture de serveur de périphérie par McMahan et Al [15]. Dans le contexte de la mise à jour des modèles linguistiques sur les téléphones mobiles ; il existe de nombreux appareils périphériques mobiles contenant des données privées.

Pour mettre à jour les modèles de prédiction dans le système Gboard qui est le système de clavier de Google pour l'auto-complétions des mots, les chercheurs de Google ont développé un système d'apprentissage fédéré pour mettre à jour périodiquement un modèle collectif. Les utilisateurs du système reçoivent une requête suggérée.

Le modèle de prédiction de mots dans Gboard marque une amélioration basée non seulement sur les données accumulées d'un seul téléphone mobile, mais sur tous les téléphones via une technique connue sous le nom de « Moyenne Fédéré » (FedAvg).

La Moyenne Fédéré ne nécessite aucun déplacement de données d'un périphérique vers un emplacement central. Bien loin de cela, avec l'apprentissage fédéré, le modèle sur chaque appareil mobile, qu'il soit une tablette ou un smartphone, est crypté puis envoyé au cloud. Tous les modèles cryptés sont agrégés dans un modèle global, de manière que le serveur du cloud ignore les données de chaque appareil.

Le modèle mis à jour est ensuite déployé sur tous les appareils individuels. Durant ce processus, les données des utilisateurs sur chaque appareil ne sont pas révélées ni aux autres, ni au serveur du cloud [14]. Il existe plusieurs catégories d'apprentissage fédéré, ces derniers vont être expliqué davantage dans le prochain paragraphe.

4.3. Les catégories de l'apprentissage fédéré :

Nous répartissons l'apprentissage fédéré en apprentissage fédéré horizontal (HFL), apprentissage fédéré vertical (VFL) et enfin apprentissage fédéré par transfert (FTL) et ceci selon la façon dont les données sont réparties entre différentes parties dans les espaces de fonctionnalités et d'échantillons [14].

4.3.1. *Apprentissage fédéré horizontal (HFL) :*

L'apprentissage horizontal HFL fait référence au cas où les participants à l'apprentissage partagent des caractéristiques de données qui se chevauchent, c'est-à-dire que les caractéristiques des données sont alignées entre les participants mais diffèrent dans les échantillons de données. Par exemple, lorsque deux parties représentent deux banques qui servent deux marchés régionaux différents, elles peuvent ne partager qu'une poignée de clients, mais leurs données peuvent avoir des caractéristiques très similaires, c'est- à -dire qu'avec un chevauchement limite des clients, mais un grand chevauchement dans les caractéristiques des données, les deux banques peuvent collaborer à la construction de modèles d'apprentissage automatique à travers des réseaux d'apprentissage fédéré horizontal [14]. La figure 8 décrit le fonctionnement de HFL.

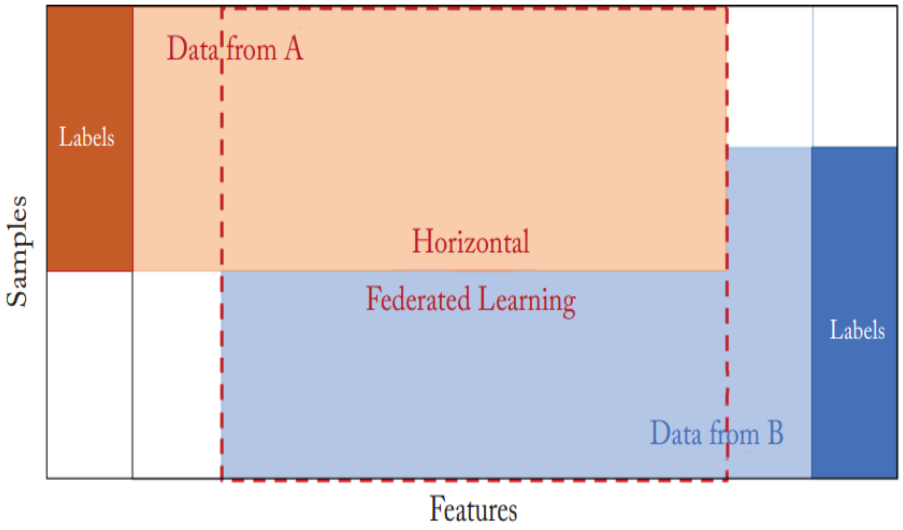

Figure 8: Représentation de l'apprentissage fédéré horizontal [14]

4.3.2. Apprentissage fédéré vertical :

Contrairement au HFL, VFL s'intéresse aux scénarios ou les participants à l'apprentissage partagent des échantillons de données qui se chevauchent mais qui diffèrent dans les caractéristiques. Cela ressemble à la situation ou les données sont partitionnées verticalement à l'intérieur d'un tableau. [14].

Par exemple, lorsque deux parties fournissent des services différents mais partagent un grand nombre d'utilisateurs (par ex une banque et une société d'e-commerce), ils peuvent collaborer sur les différents espaces de fonctionnalités qu'ils possèdent ce qui conduit à un meilleur modèle d'apprentissage automatique pour les deux. Autrement dit, avec un grand nombre d'utilisateurs en commun et peu de fonctionnalités (caractéristiques), les deux sociétés peuvent se réunir à la création de modèles ML via un apprentissage fédéré vertical [14].

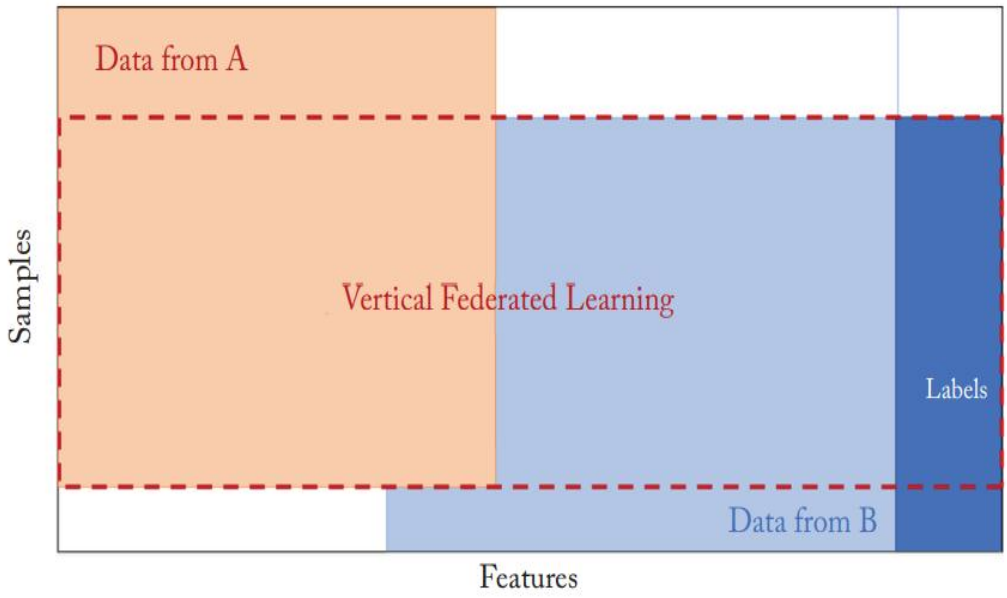

Figure 9: Représentation de l'apprentissage fédéré vertical [14]

4.3.3. *Apprentissage fédéré par transfert :*

FTL est applicable dans le cas où il n'y a aucun chevauchement que ce soit dans les données ou dans les caractéristiques [14].

Dans les scénarios où les parties participantes disposent de données très hétérogènes (incompatibilité, rare chevauchement d'échantillons et de fonctionnalités, domaine diffèrent) HFL et VFL peuvent ne pas être en mesure

de créer des modèles ML très efficaces, nous pouvons tirer parti des techniques de l'apprentissage par transfert pour combler le fossé entre les données détenues par les différentes parties [14].

L'apprentissage par transfert vise à créer des modèles ML efficaces dans un domaine cible aux ressources rares en exploitant ou en transférant les connaissances acquises à partir d'un domaine source riche en ressources, ce qui correspond naturellement au cadre d'apprentissage fédéré ou les parties appartiennent généralement à des domaines différents [14].

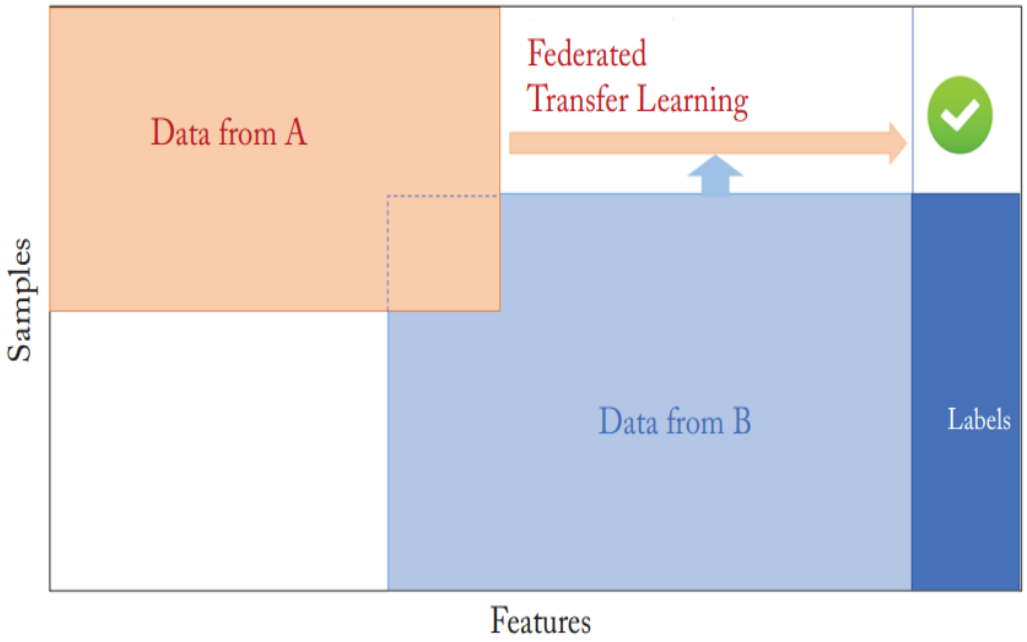

Figure 10: Représentation de l'apprentissage fédéré par transfert [14]

4.3.4. Apprentissage fédéré par renforcement :

L'apprentissage par renforcement (RL) est une branche de l'apprentissage automatique (ML) qui traite principalement les prises de décision séquentielles [Sutton et Barto, 1998]. Un problème RL consiste généralement en un environnement dynamique et d'un agent (ou des agents) qui interagissent avec lui. Ce dernier évolue une fois que l'agent sélectionne une action basée sur l'état actuel de l'environnement en présentant une récompense pour évaluer la performance de l'agent. Ce dernier cherche à atteindre un but dans l'environnement en prenant des décisions séquentielles. Les problèmes RL traditionnels peuvent être formulés comme un processus de décision Markov (MDP). L'agent doit aborder un problème de prise de décision séquentielle

pour maximiser une fonction de valeur (c'est-à-dire la somme attendue des récompenses actualisées, ou attendues) [14].

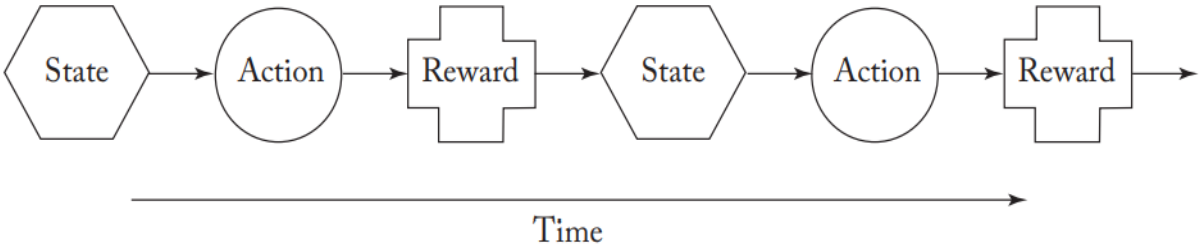

Figure 11: Représentation de l'apprentissage fédéré par renforcement [14].

4.4. Principe de l'apprentissage fédéré :

Le principe de l'apprentissage fédéré est comme suit [15] [17] [18] [19] :

En premier lieu, un serveur central crée un modèle générique, qu'il entraîne par la suite à l'aide d'une base de données publique, si cette dernière existe. Ensuite vient l'étape de la sélection des clients. Cette étape est assez importante car en fonction de la capacité des clients se déterminera la durée d'apprentissage que prendra le modèle afin d'atteindre la précision voulue : plus les clients sont compétents il faudra moins de tour. Seuls [K x C] clients seront choisis avec K le nombre total de clients et C un hyper paramètre qui définit la fraction de clients impliqués dans chaque tour.

Cette phase a connu, durant le temps, une évolution remarquable grâce aux recherches menées par les chercheurs. Plusieurs approches ont été proposées pour améliorer la sélection : la première consiste à choisir des clients aléatoirement (**VanillaFL**), la deuxième se focalise sur le critère temps (**FED-CS)**: le temps que pourra prendre un périphérique afin d'achever correctement un apprentissage ne dépassant pas une durée donnée. Par la suite une autre méthode a été proposée qui évalue les modèles en fonction de la précision des tests à l'aide d'une heuristique FL en ligne avec état (**Budget Online).**Et enfin, sélectionner un client en se basant sur ses capacités matérielles : capacité de leur RAM, CPU, mémoire… ainsi qu'à la durée d'apprentissage (**FED-MCCS).**Toutes ces méthodes vont être détaillées dans les prochains paragraphes.

Enfin, le serveur distribuera les paramètres généraux du modèle aux périphériques (clients) sélectionnés précédemment dans le but de débuter l'apprentissage.

A l'achèvement de l'apprentissage, chaque client envoie les paramètres mis à jour au serveur qui, à son tour, récoltera tous ces paramètres et les agrège pour donner une version améliorée au modèle précédent. Les étapes de la sélection des clients, distribution et l'agrégation seront répétées le nombre de fois nécessaire pour atteindre la précision désirée.

4.5. Optimisation fédérée [15]:

Nous nous référons au problème d'optimisation implicite dans l'apprentissage fédéré en tant qu'optimisation fédérée, établissant une connexion (et un contraste) avec l'optimisation distribuée. L'optimisation fédérée possède plusieurs propriétés clés qui le différencie d'un problème d'optimisation distribuée typique [15] :

- **NON-IID :** les données d'apprentissage sur un client donné sont généralement en fonction de l'utilisation de l'appareil mobile par un utilisateur particulier, et donc le jeu de données local de tout individu ne peut être représentatif du reste de la population.
- **Déséquilibrement :** de même, certains utilisateurs entraînent une utilisation plus importante d'un service ou d'une application que d'autres. Ce qui cause la variété des quantités de données d'apprentissage locales.
- **Massivement distribué :** le nombre de clients participants à une optimisation doit être largement important que le nombre moyen d'exemples par clients.
- **Communication limitée :** les appareils mobiles sont souvent hors-ligne ou sur des connexions lentes ou coûteuses.

Après que l'apprentissage fédéré vi le jour, plusieurs chercheurs et docteurs spécialisés dans ce domaine étudient cette approche et proposent des solutions aux inconvénients de ce dernier. Dans les prochains paragraphes on mit en lumière quelques études proposées concernant la partie de la sélection des clients.

4.6. L'Architecture client-serveur :

L'architecture la plus utilisée dans le système FL est l'architecture client-serveur, connue sous le nom d'architecture maître-ouvrier. Dans ce système, K participants (également appelés clients ou utilisateurs ou parties) avec la même structure de données forment en collaboration un modèle de machine learning

(ML) à l'aide d'un serveur (également appelé serveur de paramètres ou serveur d'agrégation ou coordinateur). Une hypothèse typique est que les participants sont honnêtes alors que le serveur est honnête mais curieux. Par conséquent, l'objectif est d'empêcher la fuite d'informations de tout participant vers le serveur.

5. Travaux dans le domaine de l'apprentissage fédéré :

5.1. Apprentissage fédéré vanilla :

5.1.1. Principe :

Le réseau FL composé d'un serveur de paramètre (PS : serveur de paramètre) et un nombre total de N clients. Le PS a pour but de créer un modèle ML grâce à la collaboration avec les clients sans accéder directement à leurs données brutes (privées). Soit

$$Dk \triangleq \{(x_{k,m}, y_{k,m})\}_{m=1}^{n_k}$$

le jeu de données local du client $k \in [N] \triangleq \{1, .., N\}$, où n_k est la taille des données, $x_{k,m}$ est le m-ieme échantillon de données d'apprentissage et $y_{k,m}$ est l'étiquette correspondante.

En premier lieu, le serveur choisit aléatoirement les candidats en utilisant l'algorithme Fed-Avg (federated averaging),puis ces derniers mettent à jour les modèles distribues par le serveur.

FedAvg est basé sur l'algorithme classique de stochastique gradient descent distribué (SGD : stochastique gradient descent). Dans SGD distribué, à chaque itération t, chaque client k maintient un modèle local ω_{t+1}^k par mise à jour locale du SGD.

Enfin, le serveur crée un nouveau modèle global en agrégeant les modèles locaux de chaque client, puis le PS diffuse le nouveau modèle $\underline{w}_{t+1}$ aux clients et les étapes ci-dessus sont répétées jusqu'à ce que certaine(s) condition(s) d'arrêt soient satisfaites.

L’algorithme du Fed-Avg [16] :

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
for batch $b \in \mathcal{B}$ **do**
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Algorithm du federated averaging [16].

5.1.2. Protocole du FedAvg :

le protocole du Fed-Avg [16] :

Protocol 1 FL. K Represents the Number of Participants in the Protocol. $C \in (0, 1]$ is a Hyperparameter Determining the Fraction of Clients Involved in Each Round

`1: Initialization` : The server first creates a generic model either randomly or pretrained using public data.
`2: Client Selection` : The server selects random $\lceil K \times C \rceil$ clients.
`3: Distribution` : The server disseminates the global model parameters to the selected clients.
`4: Update and Upload` : Selected clients use their local data to update the shared model and upload the new model parameters to the server.
`5: Aggregation` : The server performs an averaging process on the updated parameters to formulate an enhanced model.
`6:` Steps 2 till 5 are repeated until achieving a desired performance of the model.

Le protocole du Federated averaging [16].

Initialement le serveur génère un modèle générique pour une certaine tâche, puis sélectionne des clients aléatoirement dans le but de leur communiquer les paramètres du modèle. Le nombre de clients sélectionnés est égal à [K x C], ou K est le nombre total de clients, et C un hyperparamètre qui définit la fraction de client à impliquer dans chaque tour. Dans l'étape d'update and upload, chaque client entraîne le modèle en utilisant ses données locales et partage les nouveaux paramètres générés avec le serveur. Une fois que le serveur reçoit les modèles, il commence à les agréger afin d'avoir un modèle amélioré. Dans le cas où un grand nombre de modèles n'arrive pas au délai précisé, le serveur considère le tour comme annulé. Enfin, les étapes (sauf l'étape de l'initialisation) sont intégrées jusqu'à atteindre la performance de modèle souhaitée [17].

5.1.3. Avantages :

- Maximise le nombre de clients.
- Robuste aux distributions de données déséquilibrées et Non-IID.
- Choix des clients aléatoire ce qui réduit le taux calcul ainsi une faible complexité.
- Sécurisé : en effet les données de chaque client sont masquées vis-à-vis du serveur ainsi qu'aux autres clients, et les modèles uploader sont cryptés et chiffrés dans le but de garder la confidentialité.
- Diminution de nombre de calculs : en utilisant le SGD le nombre d'équation utilisé dans la mise à jour des paramètres est réduit à 1 équation/mini-lot au lieu de n équations/mini-lot de n clients.

5.1.4. Inconvénients :

- Le choix aléatoire cause le rejet de beaucoup de clients : en choisissant les clients aux hasard certain clients peuvent ne pas supporter le poids du processus et peut causer dans certains cas une augmentation au niveau du nombre de tour.
- La mise à jour du modèle par client peut avoir un coût de communication bien élevé ce qui pourra alourdir le processus de chargement.
- Faible intégrité au niveau des données clients : manque de vérification de la fiabilité des données au moment où les modèles sont chargés vers le serveur ce qui peut affecter la précision du modèle global, ainsi augmenter le nombre de tour.

- Hétérogénéité des données clients.
- Faible intégrité au niveau des performances des modèles.

L'inconvénient majeur de cette première méthode est le choix aléatoire des candidats d'un tour. Ce choix a comme inconvénient principale de choisir les clients les moins adapté à cette tâche, mais aussi d'augmenter le temps d'apprentissage.

Dans le prochain paragraphe, une autre méthode a été proposé qui montre que le choix des candidats en se focalisant sur leur temps d'apprentissage donne de bien mieux résultats.

5.2. FedCS : sélection des clients dans l'apprentissage fédéré

5.2.1. Principe :

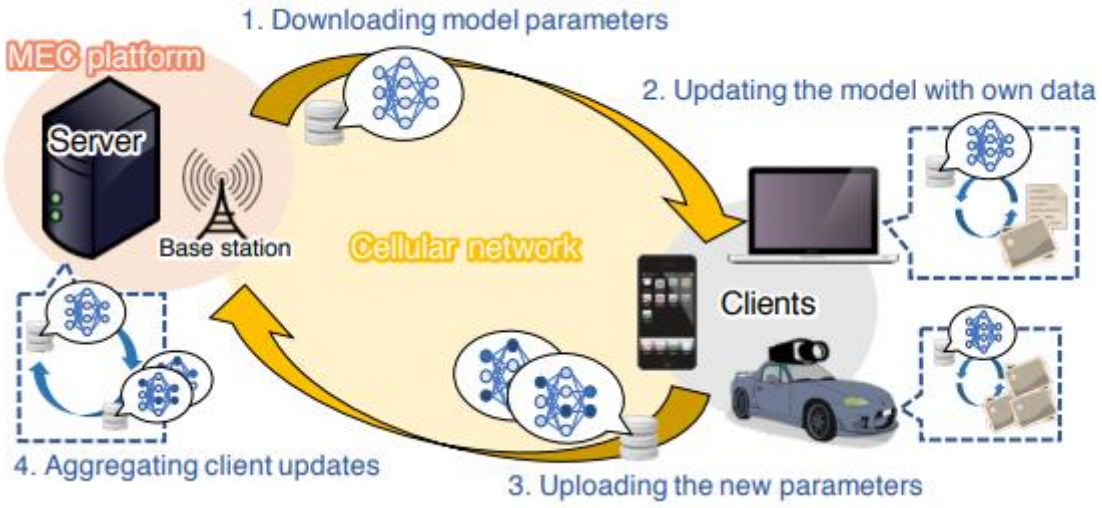

Figure 12: Illustration du principe du FedCs [18].

Comme illustré dans la figure 14, une certaine MEC (Multi-Access Edge Computing) plateforme a été considéré, qui est située dans un réseau sans fil et se compose d'un serveur et d'une station de Base BS, et qui gère les comportements du serveur et des clients dans le protocole FL. [18] suppose que les réseaux sans fil sont stables et que la quantité des ressources est gérée par l'opérateur MEC, de plus, si plusieurs clients téléchargent les paramètres du modèle simultanément, le débit pour chaque client diminue en conséquence. [18] suppose que le schéma de modulation et de codage des communications radio pour chaque client est déterminé de manière

appropriée tout en considérant son état de canal afin que le taux de perte de paquets soit négligeable. Cela conduit à un débit différent pour chaque client pour télécharger les paramètres du modèle bien que la quantité de RB (la plus petite unité de ressource de bande passante définie en LTE) soit constante [18].

5.2.2. Algorithme de l'étape de la sélection des clients [18]:

L'étape clé de cette approche est l'étape de la sélection des clients. l'explication détaillée de son algorithme sera présentée dans les prochains paragraphes.

L'objectif de cette étape est de permettre au serveur d'agréger le plus de mis à jour client que possible dans un délai déterminé. En se basant sur ce critère, l'opérateur MEC sélectionne les clients qui peuvent compléter la distribution, la mise à jour et le téléchargement planifiée dans un délai imparti. En même temps, l'opérateur programme le moment où les RB pour les téléchargements du modèle sont alloués aux clients sélectionnés pour éviter la congestion dans les bandes passantes limitées [18].

Formellement, soit $k = \{1, .., K\}$ un ensemble d'indices qui décrit K clients, et soit $k' \subseteq k$ un sous ensemble de k sélectionné aléatoirement à partir de l'étape resource request (c'est-à-dire $\left|K^{'}\right| = [\ K\ \times C]$).

$$S = [k_1, k_2, k_3, \dots, k_i, \dots, k_{|s|}], \text{ ou } k_i \in k', |s| \leq |k'|,$$

désigne une séquence d'indices des clients sélectionnés dans la sélection des clients, qui vise à être optimiser. Dans l'étape de mise à jour et de téléchargement, les clients téléchargent séquentiellement leur modèle dans l'ordre S [18].

Soit R_+ l'ensemble des réels non négatif, soit $T_{round} \in R_+$ le temps limite de chaque tour, et $T_{final} \in R_+$ le temps limite final, $T_{cs} \in R_+ et\ T_{agg} \in R_+$ le temps requis pour la sélection des clients et l'agrégation respectivement. $T_S^d \in R_+$ désigne le temps requis pour l'étape de distribution ; cette dernière dépend des clients sélectionnés S. $t_k^{ud} \in R_+$ et $t_k^{ul} \in R_+$ désigne le temps consommé par le k-ème client pour mettre à jour et télécharger les modèles, respectivement. Ces paramètres client peuvent être déterminés en fonction des informations sur les ressources notifiées dans l'étape resource request [18].

l'objectif de la sélection des clients, à savoir accepter autant de mise à jour client que possible, qui peut être atteint en maximisant le nombre de clients sélectionnés $\max_s |S|$. Pour décrire cette contrainte, [18] définit le temps écoulé estimé depuis le début de l'étape de mise à jour et de téléchargement planifiée jusqu'à ce que le k -ème client termine cette dernière procédure.

Au fur et à mesure que les clients téléchargent leurs modèle (après mise à jour) un par un, T_i^{ul} est l'accumulation de tous les temps de téléchargement requis, $t_{k_j}^{ul}$. Par conséquent, le temps de mise à jour individuel, $t_{k_j}^{ud}$, ne consomme pas T_i^{ud} tant qu'ils sont dans l'étape précédente Θ_{j-1} [18].

En résumé, la sélection des clients est formulée par le problème de maximisation suivant par rapport à S :

$$\begin{aligned} \max_{\mathbb{S}} \quad & |\mathbb{S}| \\ \text{s.t.} \quad & T_{\text{round}} \geq T_{\text{cs}} + T_{\mathbb{S}}^{\text{d}} + \Theta_{|\mathbb{S}|} + T_{\text{agg}}. \end{aligned} \tag{4}$$

Sélection du T_{round} : le paramètre le plus important dans l'algorithme est T_{round}. Si T_{round} est trop grand nous nous attendions à ce que plus de clients soient impliqués dans chaque tour. Cependant, cela réduit simultanément le nombre d'agrégations de modèles jusqu'à la date finale T_{final} [18].

l'algorithme du FedCs est représenté comme suit :

Algorithm 3 `Client Selection` **in Protocol 2**

Require: Index set of randomly selected clients $\mathbb{K}'$

Initialization $\mathbb{S} \leftarrow \{\}$, $T_{\mathbb{S}=\emptyset}^{\text{d}} \leftarrow 0$, $\Theta \leftarrow 0$
while $|\mathbb{K}'| > 0$ **do**
  $x \leftarrow \arg\max_{k \in \mathbb{K}'} \frac{1}{T_{\mathbb{S}\cup k}^{\text{d}} - T_{\mathbb{S}}^{\text{d}} + t_k^{\text{UL}} + \max\{0, t_k^{\text{UD}} - \Theta\}}$
  remove x from $\mathbb{K}'$
  $\Theta' \leftarrow \Theta + t_x^{\text{UL}} + \max\{0, t_x^{\text{UD}} - \Theta\}$
  $t \leftarrow T_{\text{cs}} + T_{\mathbb{S}\cup x}^{\text{d}} + \Theta' + T_{\text{agg}}$
  if $t < T_{\text{round}}$ **then**
    $\Theta \leftarrow \Theta'$
    add x to $\mathbb{S}$
  end if
end while
return $\mathbb{S}$

Algorithme de la sélection des clients du Fed Cs [18]

5.2.3. ***Protocol [18]:***

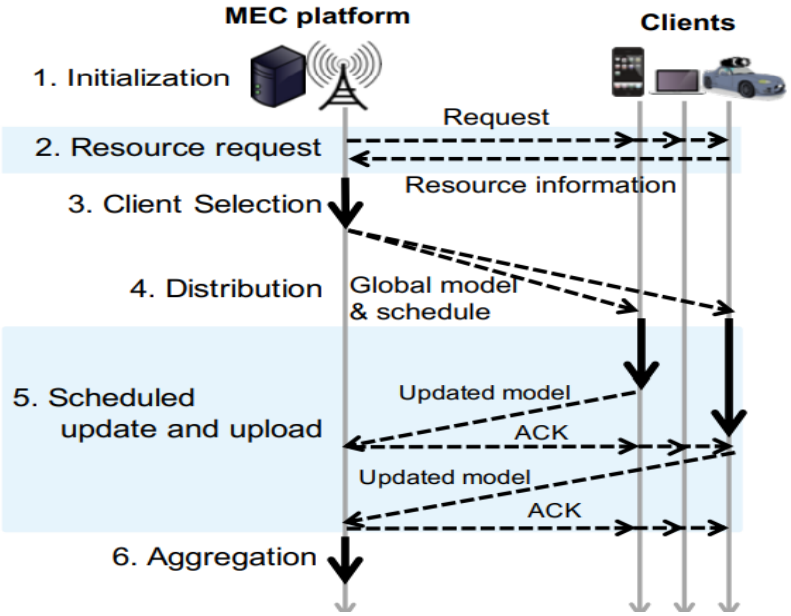

Figure 13: Vue générale sur le protocole Fed Cs [18]

La figure 16 représente une vue générale du protocole Fed-CS. Tout d'abord, la nouvelle étape de demande de ressource demande à des clients aléatoires d'informer l'opérateur MEC de leurs informations sur leurs ressources tel que l'état des canaux sans fil, les capacités de calcul, et la taille des ressources de données pertinentes pour la tâche d'apprentissage. Ensuite, l'opérateur se réfère à ces informations pour estimer le temps requis pour la distribution et l'étape de mise à jour et téléchargement du programme afin de déterminer quels clients seront sélectionnés.

Enfin, les étapes de distribution, mise à jour et téléchargement planifié et d'agrégation sont similaires à ceux de l'approche vu précédemment (FEDAvg).

La figure 14 décrit le protocole du FedCS :

Protocol 2 Federated Learning with Client Selection. K is the number of clients, and $C \in (0, 1]$ describes the fraction of random clients that receive a resource request in each round.

1: `Initialization` in Protocol 1.
2: `Resource Request`: The MEC operator asks $\lceil K \times C \rceil$ random clients to participate in the current training task. Clients who receive the request notify the operator of their resource information.
3: `Client Selection`: Using the information, the MEC operator determines which of the clients go to the subsequent steps to complete the steps within a certain deadline.
4: `Distribution`: The server distributes the parameters of the global model to the selected clients.
5: `Scheduled Update and Upload`: The clients update global models and upload the new parameters using the RBs allocated by the MEC operator.
6: `Aggregation` in Protocol 1.
7: All steps but `Initialization` are iterated for multiple rounds until the global model achieves a desired performance or the final deadline arrives.

Figure 14: Le protocole Fed Cs [18].

5.2.4. Avantages :

- Maximise le nombre de clients aptes à effectuer la tâche dans un certain délai.
- Considère les données hétérogènes.
- Temps d'apprentissage réduit considérablement.
- Sécurisé : en effet les données de chaque client sont masquées vis-à-vis du serveur ainsi qu'aux autres clients, et les modèles uploader sont cryptés et chiffrés dans le but de garder la confidentialité.
- Choisit les clients les mieux adaptés à l'apprentissage contrairement à l'approche précédente.
- Prend en compte l'état des canaux de communication.
- Diminution de charge sur le serveur grâce à la plateforme MEC.

5.2.5. Inconvénients :

- Ne considère pas l'endurance des matérielles de chaque client ce qui peut causer le drop out de ce dernier si la taille du modèle est considérablement volumineuse.
- La mise à jour du modèle par client peut avoir un coût de communication bien élevé ce qui pourra alourdir le processus de chargement.
- Faible intégrité au niveau des données clients : manque de vérification de la fiabilité des données au moment où les modèles sont chargés vers le serveur ce qui peut affecter la précision du modèle global, ainsi augmenter le nombre de tour.

Ceci étant dit, le critère du temps ne suffit pas. En effet, en se focalisant sur ce critère et en négligeant les capacités des clients, l'augmentation du nombre de tour annulés reste un inconvénient principal dans cette méthode.

L'approche suivante détermine comme critère non seulement le temps d'apprentissage du candidat, mais aussi les capacités matérielles de ce dernier.

5.3. Fed-Mccs : Modèle de sélection des clients aux multicritères pour un apprentissage fédéré optimal des IOT

5.3.1. ***Principe [17]*** :

Formellement, soit $X = \{X_1, X_2, X_3, \dots, X_k\}$ l'ensemble des clients, ayant chacun un ensemble de m pairs $\{n, l\}$, tel que n un réseau de données associées et l'étiquette $l \in \{normal, abnormal\}$. En premier lieu, on doit trouver

$X_f = \left(X_{f_1}, X_{f_2}, \dots, X_{f_i}\right)$, ou $X_f \subseteq X$ représentant les clients filtres basés sur la méthode stratifiée pour classer les clients en groupe homogène, ceux qui représente mieux l'ensemble (étape 2 protocole 3).

En deuxième lieu, maximiser le nombre de client à choisir dans l'ensemble de la sélection des clients aux multicritères (étape 4 protocole 3) représenté par

$$X_s = \left(X_{s_1}, X_{s_2}, \dots, X_{s_j}\right) \text{ ou } X_s \subseteq X_f \text{ and } j \leq [K \times C].$$

Le problème est formulé comme une maximisation à deux niveaux avec le principe du sac à dos ainsi que d'autres contraintes comme suit :

$$\max_{X_S} |X_S|$$

subject to

$$\begin{cases} \forall X_{f_{z=1}^i} \sum \text{Util}_{r\in\{\text{CPU,Memory,Energy}\}}^{X_{f_z}} < \text{Budget}_r^{X_{f_z}}[co_1] \\ \forall X_{f_{z=1}^i} \sum \left(T_d^{X_{f_z}} + \text{Util}_{r=T_{ud}}^{X_{f_z}} + T_{ul}^{X_{f_z}}\right) < T[co_2] \end{cases}$$

subject to

$$\max ER_{X_{f_{z=1}^i}} = \left[\frac{|X_{f_z}.l_A|}{|X_{f_z}.l_A| + |X_{f_z}.l_N|} \times 100\right][co_3]. \qquad (1)$$

Le but est de maximiser l'ensemble des clients sélectionnées sous 3 contraintes :

1. CO_1 : le budget limité pour l'utilisation des ressources de chaque type d'appareil de manière à ne pas décrocher. Budgets dynamiques est définie en fonction des types de ressources par type d'appareil. Un tel budget qui représente la consommation maximale de la tâche sur l'appareil, permet de compléter le processus d'apprentissage jusqu'à son achèvement.

$$Util_{r\in\{CPU,Memory,Energy\}}^{x_{fz}}$$

désigne l'utilisation prévue des ressources r pour le client x_{f_z} lors de l'entraînement d'un modèle, ou r représente le CPU, la mémoire ou l'énergie.$Budget_r^{x_{fz}}$ est le budget de la ressource par type d'appareil.

2. CO_2 : le seuil définit T n'a pas dépassé lors du téléchargement (mise à jour et téléchargement du modèle). $Util_{r=Tud}$ représente l'utilisation prévue du temps de mise à jour au moment de l'apprentissage du modèle. Tandis que $T_d^{X_c}$ et $T_{ul}^{X_c}$ désigne, respectivement, le temps requis pour le download et upload de modèle par le client X_c.
3. CO_3 : la sélection des clients en fonction de leur taux d'évènement. Les clients ont un ensemble de données Non-IID, ou ce dernier est généré en fonction de l'utilisation de chaque client de l'application liée au modèle ML. Par conséquent, certains clients peuvent produire des modèles avec des classifications déséquilibrées, avec la majorité des échantillons appartenant à une seule classe. On note ER le taux d'événement de

l'ensemble de données client, montrant la représentation des classes minoritaires, alors que $X_{f_z}.l_A$ et $X_{f_z}.l_N$ représentent, respectivement, les échantillons abnormal et normal du client X_{f_z}.

5.3.2. L'heuristique basée sur l'algorithme Greedy pour la sélection des clients [17]:

Apres avoir filtré les clients (étape 2 protocole 3), le serveur demande à chaque client dans l'ensemble filtre X_f les informations de ses ressources. Cela inclut pour chaque taille de données ses ressources utilisées, principalement le temps d'apprentissage/mise à jour, CPU, mémoire et Énergie désignées par 〖Util〗 _r^(X_c)). Après avoir reçu les réponses, le serveur lance l'algorithme de sélection des clients. Cette approche se focalise sur les clients eux-mêmes pour assurer la plus grande uniformité de distribution. [17] donne la priorité aux clients ayant le taux d'événement le plus élevé, ce qui conduit à moins de biais, même s'ils ont le plus grand nombre d'échantillons anormal. Par exemple, si un client A possède 4000 échantillons, parmi eux 70 sont abnormal, et un autre client B ayant 200 échantillons parmi lesquels seuls 50 sont abnormal, [17] donne la priorité au client B vu que le taux d'événement de ses données est égal à 25% ainsi plus important que celui de client À qui est égal à 1,75%.

Ensuite, [17] maximise le nombre de clients à sélectionner en fonction de leurs ressources. D'où éviter d'opter pour les clients incapables de terminer la phase d'apprentissage ou entraîner des réponses tardives. La méthode proposée RUPred-LR est une méthode basée sur la régression linéaire. Elle estime l'utilisation des ressources pour le prochain cycle d'apprentissage pour un client donné selon l'historique des ressources utilisées par le passé.

Chaque client sélectionné devrait être en mesure d'effectuer la tâche de FL avec un budget de ressource fixe, pour éviter les appareils surchargés.

la méthode estime le temps nécessaire pour télécharger, mettre à jour et charger le modèle. il est capable à terminer le processus dans le seuil défini T seront sélectionnés, ce qui maximise le bénéfice attendu.

l'algorithme du FedMccs est représenté comme suit :

Algorithm 1 Multicriteria Client Selection in Protocol 3

Input: $\{X_f : \forall X_{f_{z=1}^{i}}, \exists(|l_N|, |l_A|, and\ History\_X_{f_z})\}$, where:

- $History\_X_{f_z} = \bigcup_{i=1}^{n}\{x_i, y_i = Util_r\}$
- x_i is the data set size previously trained by X_z
- $Util_r$ is the resource utilization to train x_i data samples, where $r \in \{CPU, Memory, Energy, T_{ud}^{X_z}\}$
- n is the size of historical resource data set collected during FL participation of X_z

Output: The set of selected clients X_S

1: Initialize $X_S = \varnothing$
2: **while** $X_f \neq \varnothing$ *and* $|X_S| \neq \lceil K \times C \rceil$ **do**
3: $\quad X_{f_z} \leftarrow \text{argmax}_{X_{f_k} \in X_f}\left[\frac{|X_{f_k}.l_A|}{|X_{f_k}.l_A|+|X_{f_k}.l_N|} \times 100\right]$
4: $\quad$ remove X_{f_z} from X_f
5: $\quad$ **if** `sufficientResources`$(History\_X_{f_z})$ **then**
6: $\quad\quad$ add X_{f_z} to X_S
7: $\quad$ **end if**
8: **end while**
9: Return the set X_S

`sufficientResources`$(History\_X_{f_z})$
$(Util_{CPU} < Budget_{CPU}^{X_{f_z}}$ &&
$Util_{Memory} < Budget_{Memory}^{X_{f_z}}$ &&
$Util_{Energy} < Budget_{Energy}^{X_{f_z}}$ &&
$[T_d + Util_{T_{ud}} + T_{ul}] < T)$? *true*:*false*;

Algorithme de la sélection des clients du FedMccs [17].

5.3.3. RUPred-LR—prédiction de l'utilisation des ressources basée sur la régression linéaire [17]:

Selon l'historique des clients lors des derniers tour de FL, le serveur estime une fonction de prédiction pour chacun des temps d'apprentissage, CPU, mémoire et énergie.

Cette fonction présentée en (3) montre une relation linéaire entre la variable d'entrée x et la variable de sortie y_r. [17] note x la taille de l'ensemble de données du client et par y_r l'utilisation des ressources de l'appareil $Util_r$, ou r représente le temps d'apprentissage, CPU, mémoire ou l'énergie.

$$\text{Util}_{r\in\{T_{ud},\text{CPU},\text{Memory},\text{Energy}\}} = y_r = \alpha_r x + \beta_r \qquad (3)$$

Ou α_r et β_r sont les coefficients de régression du modèle.

L'objectif est d'obtenir les coefficients de régression en minimisant les résidus conduisant à la ligne du meilleur ajustement. Une des méthodes courantes de minimisation résiduelle est la régression des moindres carrés, qui trouve α_r et β_r avec des valeurs les plus minimales que possible de la somme des écarts au carré sur les n enregistrements.

5.3.4. ***Protocole [17]:***

Protocol 3 FL With Multicriteria Client Selection. K Represents the Number of Participants in the Protocol. $C \in (0, 1]$ is a Hyperparameter Determining the Fraction of Clients to Select, After Being Filtered Based on Their Metadata, and After Analyzing Their Resources

1: `Initialization` in Protocol 1.
2: `Client Filtering` : The server applies Stratified-based filtering to select clients according to their metadata, avoiding communications with irrelevant clients.
3: `Resource Request` : The server requests resource information from the filtered clients.
4: `Multicriteria Client Selection` : Based on the clients responses, the server uses Multicriteria selection approach to determine a maximum of $\lceil K \times C \rceil$ clients to participate in the remaining steps.
5: `Distribution` : The server disseminates the global model parameters to the selected clients.
6: `Update and Upload` in Protocol 1.
7: `Aggregation` : The server averages the parameters, when more than 70% of the requested updates are received.
8: All steps but `Initialization` are iterated as in Protocol 2.

Figure 15: Représentation le protocole du FedMccs [17].

Les étapes 1 5 6 7 8 sont les mêmes que les approches précédentes.

a) Client filtering : plutôt que de sélectionner des clients entièrement au hasard, [17] opte pour l'échantillonnage stratifié, qui classe la population en des sous groupes appelés strates. Ce dernier regroupe des clients homogènes partageant des caractéristiques similaires. Les métadonnées sur les clients sont déjà stockées sur le serveur ou bien partagées avec ce dernier. Cette approche affecte les clients dans les strates basées sur les métadonnées de leur région.

b) Resource request : uniquement à partir des clients filtres, le serveur demande des informations sur leurs ressources (par exemple la taille des données dont ils disposent, l'historique des données dans le tour précédent).
c) Multicriteria client selection : le serveur analyse les réponses des clients pour sélectionner le meilleur ensemble capable de participer aux prochains tours d'apprentissage.

5.3.5. Avantages :

- Minimise la complexité en minimisant le nombre de calcul à faire que sur les clients sélectionnés en se basant sur leur taux d'évènement.
- Maximise le nombre de clients en sélectionnant les meilleurs clients aptes à réaliser le cycle d'apprentissage en se basant sur la capacité de leurs hardwares ainsi qu'à leur temps d'exécution.
- Homogénéité des données.
- Minimise le nombre de tours comparé aux approches précédentes.
- Minimise le nombre de drop outs (clients rejetés).

5.3.6. Inconvénients :

- Manque d'intégrité par rapport à la performance des modèles mis à jour par les clients ce qui pourra affecter la précision du modèle global.
- Manque d'intégrité vis-à-vis des informations sur les ressources des clients.
- Le serveur peut lâcher à tout moment à cause de la charge pesé sur lui. En effet, le serveur a pour rôle d'envoyer les demandes, analyser les réponses, sélectionner les clients et créer des groupes…
- Le nombre de calculs est important car les calculs sont effectués pour chaque client de chaque tour, tout en sachant que les clients diffèrent d'un tour à un autre.

5.4. Budgeted Online : Sélection des clients IOT participants au FL

5.4.1. Le problème du secrétaire [19] :

Le problème du secrétaire, également connu sous le nom du problème de mariage, problème de dot … est une classe de problèmes de décision d'arrêt optimal. Le problème de secrétaire a été introduit la première fois par Martin Gardner en 1960. Ce problème classique se concentre sur la sélection d'une secrétaire parmi un certain nombre de candidats respectant les règles suivantes [19] :

1. Le nombre N de candidats est connu.
2. Un seul candidat doit être choisi.
3. Les candidats sont interviewés dans un ordre aléatoire.
4. Chaque candidat doit être accepté ou rejeté avant l'entretien suivant (sans possibilité de rappel plus tard).
5. Les candidats sont classés du meilleur au pire et la décision d'accepter ou de rejeter un candidat dépend des rangs relatifs des candidats interviewés jusqu'à présent.
6. Le problème consiste à maximiser la probabilité de sélection du meilleur candidat.

La solution du problème du secrétaire est pour un nombre entier $1 \leq \alpha < N$, rejeter les premiers α candidats puis sélectionner le premier candidat avec le meilleur classement que les candidats observés. Le but est de trouver α optimal qui maximise la probabilité de sélectionner le meilleur candidat. En fait, cela prouve que la valeur optimale de α est de 0.367879 avec une probabilité de (1/e) [19].

5.4.2. Principe[19] :

Le problème qu'aborde cette approche est de sélectionner le meilleur ensemble de clients qui fournit une plus grande précision de test lors de l'apprentissage du modèle global à l'aide de leur ensemble de données local [19].

Ce problème est similaire au fameux problème du secrétaire, qui vise à maximiser la probabilité de sélectionner le maximum d'éléments possibles d'une séquence ordonnée au hasard. Le problème du secrétaire est formulé comme un problème de programmation linéaire comme suit [19] :

$$\max \quad \frac{1}{N} \cdot \sum_{1=1}^{N} iP_i$$

$$\text{s.t.} \quad \forall 1 \leq i \leq N \quad i \cdot P_i \leq 1 - \sum_{j=1}^{i-1} P_j$$

$$\forall 1 \leq i \leq N \quad P_i \geq 0.$$

Dans le problème de secrétaire traditionnel, la fonction objective vise à maximiser la probabilité de choisir le meilleur candidat. Cependant, au lieu de sélectionner un seul candidat, dans ce problème, R éléments vont être sélectionnés [19].

L'heuristique proposée identifie la meilleure précision parmi les premiers candidats disponibles, cette précision est ensuite utilisée comme seuil pour accepter ou rejeter des candidats qui seront disponibles plus tard. L'heuristique accepte les paramètres N, R, r_1, r_2, K, E et s'exécute en 3 étapes toutes les δ unités de temps pour mettre à jour le modèle global [19].

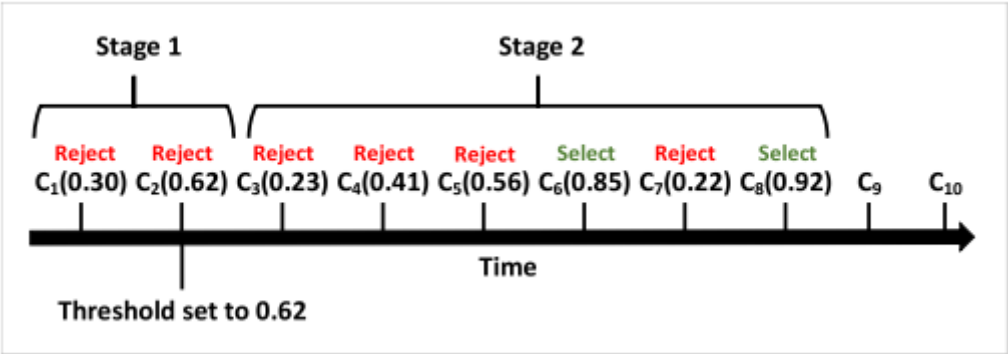

Figure 16: Exemple d'application du Budget online [19].

Dans la première étape (algorithme 1 lignes 2 à 11 [19]), la valeur de α^* est calculée en se basant sur la valeur de r_1 et r_2 (7). Les premiers clients candidats α^* qui sont disponibles sont ensuite testés pour déterminer la meilleure précision de test. Cependant, aucun de ces clients n'est accepté. Chaque fois qu'un client devient disponible, le serveur initialise le modèle global et communique avec ce client afin d'évaluer la précision de son test [19].

Les tests sont effectués en envoyant les paramètres du modèle global initialisé depuis le serveur au client pour un tour de communication afin que ce dernier entraîne le modèle à l'aide de son ensemble de données local, ensuite le client renvoie les paramètres mis à jour au serveur. Le serveur évalue les paramètres reçus (c'est-à-dire qu'aucune moyenne n'est appliquée puisqu'un

seul candidat est impliqué) en utilisant les données de test définies pour déterminer la précision du test du client. Après avoir tester α^* candidats, le serveur choisit la meilleure précision à utiliser comme seuil dans la 2eme étape [19].

Dans la 2eme étape (Algorithme 1 lignes 12 à 27 [19]), chaque fois qu'un client devient disponible, il est testé de la même façon expliquée dans l'étape précédente. Ensuite le serveur accepte le client uniquement si la précision de ce dernier est supérieure à la précision trouvée précédemment. Néanmoins, si le nombre de clients disponibles est inférieur au nombre de candidats requis (R), alors le serveur n'a d'autres choix que de sélectionner le reste des candidats. Dans le pire scénario, le candidat avec la meilleure précision est trouvé lors de première étape. Par conséquent, tous les clients rencontrés à la deuxième étape sont rejetés pour avoir une précision inférieure à la meilleure trouvée dans la première étape. Par la suite, le serveur est forcé d'accepter tous les clients trouvés durant cette étape. En effet, dans le pire scénario, l'heuristique proposée se comporte de manière similaire à l'algorithme aléatoire [19].

Dans la troisième étape (Algorithme 1 lignes 28 à 32 [19]), une fois les meilleurs clients sont identifiés, le modèle global est formé en utilisant ces derniers pour les K tours de communications [19].

5.4.3. ***Avantages :***

- Vérifie l'intégrité des modèles des clients, en calculant la précision de test de chaque modèle.
- Minimise le nombre de tours dans le meilleur scénario.
- Sécurisé au niveau du cloud : cette approche a intégré une alarme qui détecte les intrusions dans le cloud.
- Sécurisé au niveau des clients.
- Maximise le nombre de clients.

5.4.4. Inconvénients :

- Cette approche n'est pas performante à tous les coups : en effet la probabilité que le pire scénario se produise est bien plus élevée que celle du meilleur.
- Les données sont hétérogènes.
- Cette approche n'est pas applicable que dans l'état en ligne du FL.

- Ne se base que sur le critère de précision.
- Cette approche utilise les mêmes candidats durant tous les tours de communication (c'est-à-dire durant tout l'apprentissage).

6. D'autres travaux dans l'apprentissage fédéré :

Les travaux de recherche récents sur l'apprentissage fédéré se focalisent principalement sur l'amélioration de l'aspect de la sécurité et des défis statistiques par Yang et al., et Mancuso et al. Cheng et al. a proposé SecureBoost dans le cadre de l'apprentissage fédéré vertical, qui est un nouveau système de préservation de la vie privée de renforcement d'arbre (tree-boosting system). SecureBoost offre le même niveau de précision que l'approche sans préservation de la vie privée. Il est théoriquement prouvé que le framework SecureBoost est aussi précis que d'autres algorithmes non fédérés de tree-boosting de gradient qui repose sur des ensembles de données centralises [14].

Liu et al. présente un cadre flexible d'apprentissage par transfert fédéré qui peut être efficacement adapte à diverses taches ML multi-parties sécurisées. Dans ce cadre, la fédération permet le partage de connaissances sans compromettre la vie privée des utilisateurs et permet le transfert de connaissances complémentaires dans le réseau via l'apprentissage par transfert. Par conséquent, une partie du domaine cible peut créer des modèles plus flexibles et plus puissant en tirant parti des étiquettes riches d'une partie du domaine source [14].

Dans un système d'apprentissage fédéré, les parties participantes peuvent être honnêtes, semi honnêtes ou malicieuses. Lorsqu'une partie est malicieuse, il est possible qu'un modèle altère ses données dans l'apprentissage. La probabilité qu'un agent malveillant initie une attaque sur un modèle dans le FL dans le but de l'empoisonner est discuté dans Bhagoji et al. Un certain nombre de stratégies pour effectuer une attaque d'empoisonnement du modèle ont été étudiés. Il a été montré que même dans un environnement contraint, l'adversaire peut effectuer des attaques sur le modèle tout en maintenant sa furtivité. Le travail de Bhagoji et al. révèle la vulnérabilité des dispositifs du FL et prône la nécessite de développer des stratégies de défenses efficaces [14].

La réexamination des modèles ML existant dans les paramètres de l'apprentissage fédéré est devenu une nouvelle direction de recherche. Par exemple, combiner l'apprentissage fédéré avec l'apprentissage par renforcement a été étudié dans Zhuo et al. ,ou Gaussian differentials sur l'information partagée entre les agents lors de la mise à jour de leurs modèles locaux ont été appliques pour protéger la confidentialité des données du modèle. Il a été démontré que le modèle d'apprentissage fédéré par renforcement propose et fonctionne proches des lignes de base qui prennent directement toutes les informations conjointes en entrée …[14].

7. Discussion :

Apres avoir présenter les approches les plus pertinentes existantes dans la littérature tel que le vanillaFL, Fed-Cs et le Fed-Mccs, nous nous focalisons sur ces quelques points à développer dans l'approche FedMccs, qui sont décrits dans les points suivants :

- Dans la première contrainte de la formule, ils définissent une borne supérieure pour chaque ressource, ex. CPU, mémoire, énergie, de chaque type d'appareil. En effet, ils estiment que c'est le meilleur moyen afin d'éviter le drop out des clients. Sur ce fait, leur approche n'a pour but que de maximiser le nombre de candidats participant durant un tour et non pas sélectionner les candidats les plus performants.
- L'auteur a opté pour un algorithme de prédiction basé sur la régression linéaire. Parmi les étapes de cet algorithme**,** c.à.d. que pour chaque critère de chaque appareil, le serveur estime une fonction de prédiction en se basant sur l'historique de chaque client. La fonction montre une relation linéaire entre la taille de l'ensemble, de donnée de chaque client avec l'utilisation des ressources de l'appareil. En estimant la complexité de cet algorithme, celle-ci reste tout aussi importante avec le nombre de fonction à estimer et à résoudre par la suite, même si le nombre de candidats diminue grâce à l'étape de la création des groupes homogènes.

- Dans l'approche FedMccs, l'auteur ne prend en compte que les compétences matérielles du client tel que le CPU, GPU, mémoire et énergie, ainsi que la durée d'apprentissage durant lequel le client télécharge le modèle, l'entraîne en utilisant ses propres données et enfin charge ce dernier sur le serveur. Par conséquent, plusieurs critères ont été négligé, tel que l'état de la communication réseau entre un client et le serveur par exemple.

8. Conclusion :

Dans cet état de l'art, nous avons mis en lumière les recherches les plus pertinent sur la sélection des clients.

Parmi les méthodes citées précédemment, nous sommes intéressés, pour notre conception, sur la méthode FedMccs. Cette dernière est une amélioration des méthodes vanillaFL et Fedcs. En effet, cette méthode englobe les critères qui nous semblent les plus importants.

Dans le prochain chapitre, nous modélisons notre proposition en commençant par la formulation du problème, suivie des algorithmes utilisés dans le but d'apporter une amélioration au Fed-Mccs.

Chapitre III: Approche proposée

1. Introduction :

Le but principal de ce travail est d'optimiser la sélection des clients dans l'apprentissage fédéré.

Dans notre proposition, les clients utilisés sont des clients non étiquetés, c.-à-d. non catégorisés, pour cela les algorithmes d'apprentissage automatique non supervisés sont les mieux appropriés dans notre cas . Dans cette catégorie, il existe plusieurs algorithmes de regroupement ou de clustering tels que K-Means, Hierarchical clustering…

Nous avons utilisé l'algorithme du K-means pour les raisons suivantes : Il est facile d'identifier des groupes de données inconnus à partir d'ensembles de données complexes, Il convient à un grand nombre d'ensembles de données, Il peut également produire des clusters plus élevés, pour notre cas plus le nombre de critères est important, plus l'algorithme est précis. De plus, La segmentation en K-Means est linéaire en nombre d'objets de données, ce qui augmente le temps d'exécution. Il ne faut pas plus de temps pour classer des caractéristiques similaires dans des données telles que des algorithmes hiérarchiques.

Notre étude bibliographique, synthétisée dans le chapitre précèdent, nous a permis de cerner les critères à prendre en considération dans cette étude. En effet, la source de notre inspiration Fed-Mccs englobe les critères les plus influents de cette étape, nous citons dans ce cas : le CPU, la capacité mémoire, la frequence et le temps d'apprentissage.

Ce chapitre est la représentation de notre proposition qui aura comme résultat les clients les mieux adéquats afin d'entrainer au mieux les modèles d'apprentissage automatique dans le domaine de la domotique et avoir de bonnes précisions dans l'apprentissage fédéré.

Nous présentons, dans ce qui suit, le principe général de notre approche.

2. Principe générale de l'approche proposée :

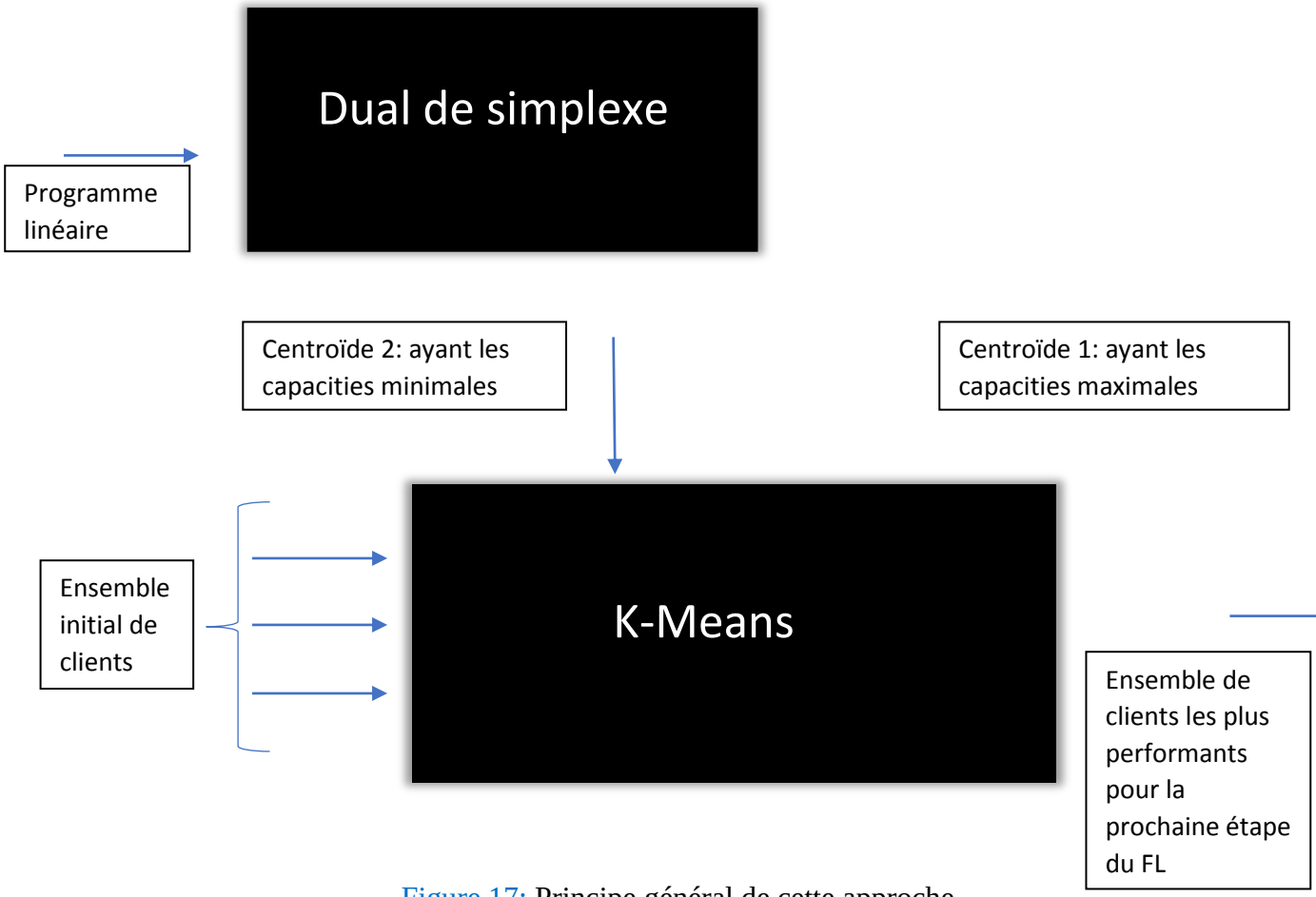

Figure 17: Principe général de cette approche.

La figure 21 représente un schéma global de l'approche que nous avons proposé. A partir d'un ensemble de client initial, nous allons extraire les clients les plus pertinents. Pour notre cas, nous avons choisi un algorithme d'apprentissage automatique le K-means.

Apres avoir défini le nombre de cluster souhaite, pour notre cas 2, ainsi que les centroïdes de ces clusters, le K-means va calculer la distance de similarité entre les clients et les centroïdes pour qu'au final classer les clients, selon la distance minimale, dans la casse qui leur convient.

Pour le choix des centroïdes : nous avons sélectionner aléatoirement le centroïde minimal, contrairement au centroïde maximal qui a été calcule en utilisant le dual de simplexe.

Remarque 1 : les capacités du centroïdes 1 sont calculées en utilisant le dual de simplexe.

Remarque 2 : les capacités du centroïdes 2 sont choisis aléatoirement parmi les clients ayant de mauvaises capacités.

3. Pseudo code :

```
4. Fonction selection des clients (Message m)
5. Cmax(CPU, temps, Ram, frequence) ;
6. Cmin (CPU, temps, Ram, frequence) ;
7.
8.     CPU <-getCPU(m) ;
9.     Temps <-getTemps(m) ;   // extraction des valeurs
10.    RAM<-getTRam(m) ;
11.    Frequence= getfrequence(m) ;
12.
13.    Client (CPU, Temps, Ram, Frequence) // affecter les
   valeurs extraites au client
14.
15.        Dmax<-cal_D(Cmax,client) ; // distance entre cmax et
   client
16.         Dmin<-cal_D(Cmin,client) ;
17.
18.    If Dmax < Dmin then
19.       Cluster 1 <-client ;
20.
21.    // cluster 1 et 2 sont des variables globales
22.
23.    Else Cluster 2 <-client ;
24.      Fin Fonction
```

4. L'algorithme utilisée :

Dans le chapitre précèdent, nous avons introduit quelques notions de base sur l'apprentissage automatique ainsi que les différents types de ce dernier. Parmi ces types il existe l'apprentissage automatique non supervisé qui vise principalement la classification des données non étiquetées en d'autres termes le clustering.

En premier temps, dans la classification non supervisée, aucune classe / cluster n'a été créé, ce qui veut dire qu'on se trouve avec des données dispersées et non étiquetées.

Parmi les algorithmes de classification non supervisée, le k-means a pour but de classifier ces données en cluster, regroupant les données les plus similaire, en calculant la distance et en sélectionnant la plus minimal entre une donnée et un centre d'un cluster, en créant k cluster (classe). Chaque cluster est défini par un centroïde, ce dernier varie durant l'exécution de l'algorithme.

Nous résumons les étapes de l'algorithme dans les points suivants :

i. Définition du K : le nombre de clusters est définit au départ.
ii. Dans un premier temps, un ensemble de k centres est choisi aléatoirement parmi l'ensemble des données.
iii. Les k clusters sont formés en calculant la distance entre une donnée avec les K centres. La donnée est affiliée au cluster ayant le centre le plus proche de cette dernière : choix de la distance la plus petite. Comme métrique de distance nous avons utilisé la ***distance euclidienne***, cette distance est la plus utilisée pour sa simplicité et son efficacité. dans notre cas elle a donné de bons résultats (présentes dans le chapitre suivant).
iv. Le centre de chaque cluster est recalculé afin de définir le nouveau centre.
v. Les étapes précédentes sont en boucle jusqu'à immobilité des centres. Dans notre cas nous avons fixé le nombre de tours à 1, car nos centroïdes sont calculés, prédéfinis.

En ce qui concerne le choix des centroïdes, nous définissons deux centroïdes : centroïde Max et centroïde Min.

Pour le centroïde Min nous l'avons sélectionné aléatoirement, et pour le centroïde Max, nous avons choisi de le calculer en utilisant la méthode du dual de simplexe, cette méthode requière un programme linéaire qui est définit dans ce qui suit.

5. Formulation du problème :

Dans cette section, nous considérons notre problème comme un problème de sac à dos, c.-à-d. qu'on a un nombre maximal de clients à sélectionner et les valeurs (capacités) de ces derniers doivent être maximisés. Pour cela, nous utilisons la programmation linéaire PL.

Le PL se compose d'une fonction objective qui exprime la maximisation des capacités des clients, suivie de contraintes reliées à ces capacités. La formulation du problème devient comme suit :

$$\max z = C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 \quad (1)$$

$$C_{min} \leq x_1 \leq C_{max} \quad (2)$$

$$x_2 \leq T \quad (3)$$

$$x_3 \geq Ta_{min} \quad (4)$$

$$F_{min} \leq x_4 \leq F_{max} \quad (5)$$

$$x_1, x_2, x_3, x_4 \geq 0 \quad (6)$$

- (1) représente une fonction linéaire visant à maximiser un certain z. Ce dernier désigne la performance d'un client en fonction de ces capacités matérielles tel que le CPU et la mémoire ainsi qu'en fonction du temps d'apprentissage que prend chaque client et aussi en fonction de l'état du canal de transmission. Ces capacités sont multipliées par des coefficients : C_1, C_2, C_3, C_4 , qui montrent à leur tour, l'importance d'un critère : plus le coefficient est grand, plus le critère est important.
- (2) désigne la première contrainte qui concerne la capacité CPU (x_1). On a opté pour une borne supérieure (C_{max}) afin de limiter le drop out des clients. Dans cette proposition, nous avons rajouté une borne inferieur (C_{min}) dans le but d'ignorer les clients faibles et de sélectionner ceux qui ont une bonne capacité de calcule. Ce choix reflète le fait que les modèle ont une complexité de calcule assez importante : Plus la taille des données d'apprentissage est volumineuse, plus la complexité de calcules est importante.
- (3) correspond au critère du temps d'apprentissage (x_2) qui peut être effectuer durant un tour.
    - T_{ap} : temps d'apprentissage.
    - T_t : temps de téléchargement du modèle.
    - T_{en} : temps d'entrainement du modèle.
    - T_{ch} : temps de chargement de la nouvelle version du modèle sur le serveur.

    $T_{ap} = T_t + T_{en} + T_{ch}$.

  Ce critère a une grande importance car il définit si un client peut terminer les tâches dans un délai choisi T mais aussi définit le temps que prendra le modèle global pour atteindre une certaine précision avec la diminution du temps.
- (4) cette contrainte désigne le critère de la taille de mémoire (x_3). Afin d'accomplir la tâche d'apprentissage et durant la phase de calcule, le

client doit contenir une bonne RAM. Vu la complexité des modèles d'apprentissage la qualité de la RAM est indispensable et doit avoir une capacité supérieure ou égale à un certain Ta_{min}.

- (5) cette contrainte concerne l'état du canal de communication entre le serveur et le client (x_4). Effectivement, l'état des canaux de communication ont aussi leur importance dans le FL vu que le serveur et le client interagissent entre eux durant plusieurs étapes : demande d'information, distribution, chargement du modèle du client au serveur. Pour vérifier l'état du canal, on a opté pour la bande passante, une des métriques de communication réseau. La **bande passante** (angl. *bandwidth*) est un intervalle de fréquences pour lesquelles la réponse d'un appareil est supérieure à un minimum. La bande passante est la largeur, mesurée en hertz, d'une plage de fréquence F_{max} - F_{min}. Elle peut aussi être utilisée pour décrire un signal, dans ce cas le terme désigne la différence entre la plus haute et la plus basse fréquence du signal (ce que l'on appelle aussi l'encombrement spectral).

- (6) cette contrainte est obligatoire, dans un programme linéaire les variables doivent être égales ou supérieure à 0.

6. Critères considérés:

À partir des seuls clients filtrés, le serveur demande des informations sur leurs ressources. Dans notre mémoire, et après avoir étudié les appareils IOT existants dans le domotique, nous nous sommes appuyés sur les aspects suivants, selon un ordre d'importance (du plus important au moins) :

- ***Capacite de calcul*** : les Applications prenant en charge des services nécessitant un traitement d'image et de vidéo, tels que Systèmes de détection d'intrusion, nécessitent une grande capacité de calcul (CPU, mémoire, énergie, GPU) pour exécuter divers algorithmes d'apprentissage automatique et effectuer des traitements complexes. Les services peuvent ne pas avoir des contraintes de latence extrêmement faibles; cependant, les serveurs Edge inactifs distant peuvent allouer des ressources à ces applications. De plus, certains appareils peuvent ne pas performer certaines tâches d'apprentissage si elles disposent d'une grande quantité de données, et/ou faibles ressources de calcul. Cela affecte les périphériques en provoquant une panne du système et, par conséquent, augmente le nombre de cycles de communication nécessaires pour atteindre une précision cible pour le modèle.

- ***Temps de calcul :*** En utilisant l'information, le serveur détermine lequel des clients passent aux étapes suivantes pour terminer les étapes dans un certain délai. Par exemple, lorsque certains clients ont une capacité de calcul limité, ils nécessiteront plus de temps pour mettre à jour les modèles. De plus, si les canaux sont relativement petit comparés à la taille des données envoyées, ceci entraînera un temps de mise à jour plus long.
- ***Interactions fréquentes :*** l'apprentissage fédéré nécessite une interaction fréquente entre les appareils IoT et les applications hébergées sur des serveurs. Pour cela, l'état de la bande passante doit être stable car, une interaction fréquente des informations (tels que les demandes de ressources, l'envoie des paramètres du modèle…) avec les applications correspondantes est nécessaire. Portable intelligent, les électroménagers intelligents ainsi que les appareils surveillant l'état de santé doivent charger et télécharger des modèles durant l'apprentissage fédéré.
- ***Taille de la mémoire du client :*** afin d'éviter le dysfonctionnement des appareils, nous prenons aussi comme critères la taille de leur mémoire. Notant que l'importance de cette dernière est tout aussi capitale. En effet, le CPU, durant un apprentissage, a besoin de stocker les instructions sur un composant. De plus, on a besoin d'une grande RAM .

7. Résolution du problème :

Afin de résoudre ce programme linéaire plusieurs méthodes ont été proposées dans le domaine d'optimisation combinatoire. Parmi ces dernières, nous avons opté pour la plus convenable la méthode du dual de simplexe qui permet de résoudre les programmes linéaires. Cette méthode est simple à manipuler et tout aussi efficace et rivalise avec la méthode de simplexe, c.-à-d. qu'au moment où on ne peut pas utiliser cette méthode, on opte pour le dual.

Le dual permet d'éliminer les valeurs négatives des contraintes afin de mieux trouver une valeur optimale aux variables. Contrairement à la dualité, le dual simplifie la transformation du programme linéaire durant l'étape de la recherche de la forme admissible, en d'autres termes, le dual est largement moins couteux que la dualité mais tout aussi efficace.

Etant donnée le programme linéaire ci-dessus, les contraintes de ce dernier comportent les signes $\leq$ et $\geq$, ce qui veut dire qu'on ne peut utiliser ni la méthode de simplexe ni la méthode de dualité, car ces méthodes requirent que les contraintes du PL ne contiennent que le signe$\leq$ et ne doivent pas contenir des valeurs négatives.

Initialement, nous fixons les coefficients de manière aléatoire comme suit : C1=25, C2=15, C3=10, C4=5, ces dernières ont été fixés ainsi afin de définir l'importance des critères du plus prioritaire au moins.

Ainsi que les bornes supérieurs et inférieurs des contraintes : Cmin=2.35, Cmax=5 (GHz), T=60 (s), Tamin= 4 (Go), Fmax=5 et Fmin=0,16 (GHz), ces valeurs ont été choisi parmi les clients ayant les capacités entre moyennes et élevées, notre but est de sélectionné les clients performants/moyennement performant pour maximiser le nombre de client. Le PL devient comme suit :

$$\max z = 25x_1 + 15x_2 + 10x_3 + 5x_4 \quad (1)$$

$$2.35 \leq x_1 \leq 5 \quad (2)$$

$$x_2 \leq 60 \quad (3)$$

$$x_3 \geq 4 \quad (4)$$

$$0.16 \leq x_4 \leq 5 \quad (5)$$

$$x_1, x_2, x_3, x_4 \geq 0 \quad (6)$$

Remarque : les valeurs des bornes supérieurs et inférieurs sont sélectionnées de sorte que les clients choisis font partie des classes moyens à compétents, ainsi maximiser le nombre de clients dans chaque tour. Dans le domaine de la domotique, il existe différents types d'appareils IOT avec différents niveaux de performance, on peut les classer comme suit : mauvais, moyen et bon. Dans notre proposition nous avons choisi de sélectionner les bons afin bénéficier de leurs capacités, mais aussi les moyens afin de maximiser le nombre des clients tout en évitant les inconvénients des mauvais (lenteur, tour annulé…).

Après avoir fixé les coefficients et les bornes, on cherche la forme admissible par l'algorithme de *dual de simplexe,* c.-à-d. transformer les signes $\geq$ et $\leq$ en =.

Ainsi le PL devient comme suit :

$$-z + 25\,x_1 + 15\,x_2 + 10\,x_3 + 5\,x_4 = 0$$
$$x_1 + e_1 = 5$$
$$-x_1 + e_2 = 2.35$$
$$x_2 + e_3 = 60$$
$$-x_3 + e_4 = -4$$
$$x_4 + e_5 = 5$$
$$-x_4 + e_6 = -0.16$$

Ensuite en utilisant le logiciel LP Assistant, illustré ci-dessous, nous trouvons les résultats optimaux suivants : $x_1 = 2.35,\ x_2 = 30, x_3 = 4, x_4 = 0.16$.

Ce logiciel nous aide à créer les tableaux, calculer et recalculer les valeurs du tableau. Il nous suffit de trouver le pivot. 2 algorithmes sont proposés dans ce logiciel : le simplexe et le dual de simplexe. Dans notre cas, ce logiciel nous a permis de calculer les valeurs du tableau, de plus de trouver les valeurs optimales des variables en utilisant le dual de simplexe.

Illustration du LP Assistant :

1- Initialisation du tableau :

Durant cette phase, 2 étapes sont essentielles : en premier lieu, remplir le tableau avec les valeurs du PL. en deuxième lieu, choisir la méthode à utiliser le simplexe ou le dual de simplexe.

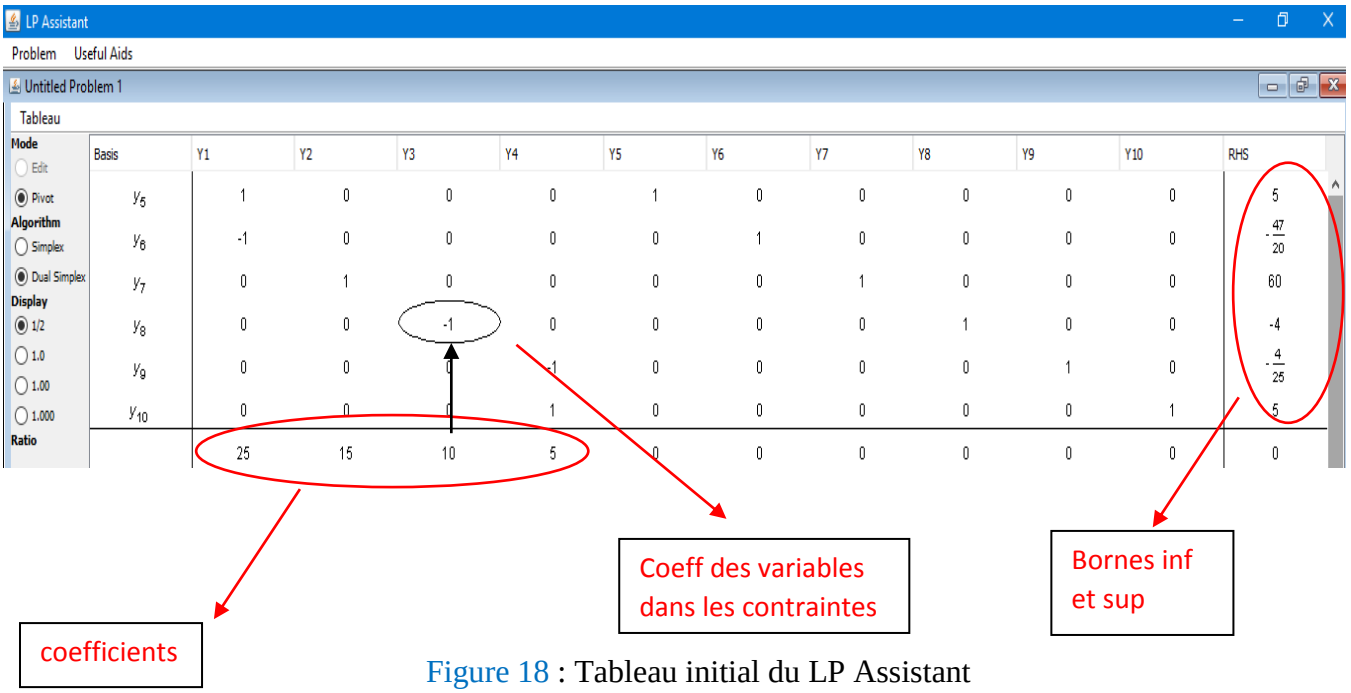

Figure 18 : Tableau initial du LP Assistant

Le but de dual de simplexe est d'éliminer les valeurs négatives. D'abord, nous sélectionnons la lignes ayant la plus petite valeur parmi les bornes inférieures et supérieures, ensuite sélectionner, dans notre cas de maximisation, le min (coefficient/Coeff des variables dans les contraintes). Le Coeff des variables dans les contraintes doit être de la même colonne que le coefficient de la fonction objective, mais aussi doit faire partie de la même ligne que la valeur minimale sélectionnée parmi les bornes. Le Coeff des variables dans les contraintes qui vérifie la condition précédente sera nommé le pivot.

Les valeurs du tableau sont recalculées selon les règles mathématiques illustrées ci-dessous jusqu'à atteindre le but de cette méthode :

A. Diviser la ligne du pivot par a_{st} (la valeur du pivot).

$$a'_{sj} = \frac{a_{sj}}{a_{st}}, 1 \leq j \leq n+m$$

$$b'_s = \frac{b_s}{a_{st}}$$

où s est l'indice de la ligne de la variable sortante, t est l'indice de la colonne de la variable entrante, et a_{st} correspond à la valeur du pivot.

B. Calculer les autres valeurs

$$a'_{ij} = a_{ij} - (a_{it} \times a_{sj})/a_{st} \qquad 1 \leq i \leq m, i \neq s$$

$$1 \leq j \leq n+m$$

$$b'_i = b_i - (a_{it} \times b_s)/a_{st} \qquad 1 \leq i \leq m, i \neq s$$

$$-z' = -z - (c_t \times b_s)/a_{st}$$

$$c'_j = c_j - (c_t \times a_{sj})/a_{st} \qquad 1 \leq j \leq n+m$$

2- Tableau final :

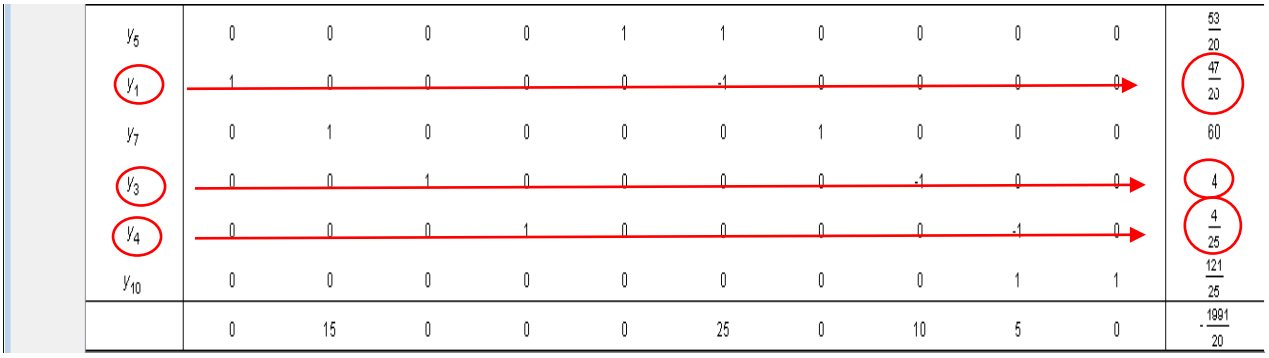

Figure 19: Tableau final du LP Assitant

Remarques :

- ***a.*** Les valeurs trouvées en utilisant le LP Assistant sont les valeurs optimales du PL. Y_1=2.35, Y_3=4, Y_4=0.16 désigné respectivement x_1, x_3, x_4.
- ***b.*** x_2= 0 par le logiciel. Mais afin d'optimiser nos recherches nous avons opté x_2=30 qui a donné de meilleurs résultats (présentés dans le chapitre suivant).
- ***c.*** Les valeurs trouvées vont être pris comme capacité du client optimal qui, par la suite, va être considéré comme centroïde du cluster des clients à sélectionner dans l'algorithme du K-means.

Apres avoir trouvé le client optimal, nous utiliserons le K-means afin de générer le cluster comportant les clients les mieux et les moyennement compétents qui participeront dans le FL.

8. Conclusion *:*

Dans ce chapitre, nous avons présenté la sélection des clients en utilisant un des algorithmes de l'apprentissage automatique qui est le K-Means.

Tout d'abord, nous avons calculé les valeurs du centroïde optimal en utilisant une méthode d'optimisation combinatoire le *dual de simplexe*, et sélectionné un centroïde qui ne vérifie aucune contrainte citée dans le PL.

Ensuite, nous avons utilisé ces derniers dans l'algorithme du K-Means qui nous donne en sortie 2 clusters : le premier comportant les clients ayant les

performances les mieux adéquat mais aussi les moyens, et le deuxième comportant les pires clients ayant de mauvaises compétences.

Le prochain chapitre consistera à présenter les résultats obtenus en comparant 3 méthodes de sélection : la sélection aléatoire, la sélection par temps et la sélection en utilisant l'algorithme du K-means.

Chapitre IV: Implémentation et simulation

1. Introduction :

Ce dernier chapitre est l'exaucement du travail élaboré dans ce mémoire. Nous présentons notre méthode de sélection des clients "sélection par k-means". Nous avons simulé cette dernière ainsi que les deux autres méthodes "sélection aléatoire" et "sélection par temps d'apprentissage" pour comparer les résultats sous le simulateur OMNET++ 5.7.

Nous commençons par la présentation de l'environnement, et procédons par la suite à la base de données utilisée ainsi qu'aux réglages de paramètres et de jeux de test effectués. Les résultats du test de notre méthode sont comparés avec les résultats des deux autres méthodes en utilisant les mêmes clients qui sont définis dans la suite de ce chapitre.

Tous les tests ont été réalisé sur Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40 GHz avec 4 Go de RAM, et un système d'exploitation Windows 10.

2. Environnement de la simulation :

Nous allons détaillés dans un bref aperçu les outils utilisés dans la réalisation de notre simulation.

2.1. **OMNET++ :**

La plateforme OMNeT++ :

OMNeT++ est une plateforme de simulation, modulaire, open source, orienté objet et à événements discrets écrit en C++. Elle offre un IDE basé sur Eclipse, un environnement d'exécution graphique, et une foule d'autres outils. Elle a été conçue pour créer des simulateurs pour les réseaux de communication, les systèmes multi processeurs, et d'autres systèmes distribués. Le développement d'**OMNeT++** a commencé en 1992 par Andras Vargas à l'université de Budapest. Actuellement, Ce projet est utilisé par des dizaines d'université pour la validation de nouveaux matériaux et logiciels, ainsi que pour l'analyse de performance et l'évaluation de protocoles de communication. L'avantage de **OMNeT ++** facilite l'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés [20].

i. L'architecture d'OMNET++ :

L'architecture d'OMNET++ est hiérarchique composé de modules. Un module peut être soit module simple ou bien un module composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque module simple correspond un fichier .cc et un fichier .h. Un module composé est construire de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un fichier .ned (ned : Network Description). La communication entre les différents modules se fait à travers les échanges de messages (.msg). Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service. Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module. La conception d'un réseau se fait dans un fichier .ned et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration (.ini). OMNeT++ génère à la fin de chaque simulation deux nouveaux fichiers omnet.vec et omnet.sca qui permettent de tracer les courbes et calculer des statistiques [20].

2.2. Les clients utilisés dans la simulation :

Dans cette simulation, nous avons créé 137 clients de test, représentés dans le tableau. les critères de ces clients sont inspirés par les appareils IOT existants dans la domotique tel que les électro-ménagers intelligents.

Tableau 1: Représentation des clients utilisés pour effectuer la comparaison entre les méthodes.

clients	CPU (GHz)	Temps d'apprentissage (s)	Mémoire RAM (Gb)	Fréquence de bande passante (GHz)
1	0.6	110	8	2.4
2	0.6	110	8	2.4
3	0.6	110	8	2.4
4	0.6	110	8	2.4
5	0.6	110	8	2.4
6	1.58	90	8	2.4

7	1.58	90	8	2.4
8	1.58	90	8	2.4
9	1.58	90	8	2.4
10	1	120	0.064	0.16
11	1	120	0.064	0.16
12	1	120	0.064	0.16
13	1	120	0.064	0.16
14	1	120	0.064	0.16
15	2.26	50	4	5
16	2.26	50	4	5
17	2.26	50	4	5
18	2.26	50	4	5
19	2.26	50	4	5
20	2.35	60	6	2.4
21	2.35	60	6	2.4
22	2.35	60	6	2.4
23	2.35	60	6	2.4
24	2.35	60	6	2.4
25	3.3	40	4	2.4
26	3.3	40	4	2.4
27	3.3	40	4	2.4
28	3.3	40	4	2.4
29	5	30	64	2.133
30	5	30	64	2.133
31	5	30	64	2.133
32	1.5	90	8	1.9
33	1.5	90	8	1.9
34	1.5	90	8	1.9
35	1.5	90	8	1.9
36	1.5	90	8	1.9
37	2.26	80	4	1.6
38	2.26	80	4	1.6
39	2.26	80	4	1.6
40	2.5	50	8	4.2
41	2.5	50	8	4.2
42	2.5	50	8	4.2
43	2.5	50	8	4.2
44	2.5	50	8	4.2
45	2.5	50	8	4.2

46	3.6	39	6	2.6
47	3.6	39	6	2.6
48	3.6	39	6	2.6
49	3.6	39	6	2.6
50	3.6	39	6	2.6
51	3.6	39	6	2.6
52	4.77	30	8	4.7
53	4.77	30	8	4.7
54	4.77	30	8	4.7
55	4.77	30	8	4.7
56	0.6	120	4	0.8
57	0.6	120	4	0.8
58	0.6	120	4	0.8
59	0.6	120	4	0.8
60	3	38	2	2.6
61	3	38	2	2.6
62	3	38	2	2.6
63	3	38	2	2.6
64	4	30	16	4.9
65	4	30	16	4.9
66	4	30	16	4.9
67	4	30	16	4.9
68	4	30	16	4.9
69	2.5	60	32	1.3
70	2.5	60	32	1.3
71	2.5	60	32	1.3
72	2.5	60	32	1.3
73	1.5	70	4	2.7
74	1.5	70	4	2.7
75	1.5	70	4	2.7
76	1.5	70	4	2.7
77	3.6	32	8	4
78	3.6	32	8	4
9	3.6	32	8	4
80	3.6	32	8	4
81	3.6	32	8	4
82	2	60	4	2.4
83	2	60	4	2.4
84	2	60	4	2.4

85	2	60	4	2.4
86	2	60	4	2.4
87	2.05	45	6	3.8
88	2.05	45	6	3.8
89	2.05	45	6	3.8
90	1.8	85	3	2.2
91	1.8	85	3	2.2
92	1.8	85	3	2.2
93	1.8	85	3	2.2
94	1.8	85	3	2.2
95	1.6	80	2	2.9
96	1.6	80	2	2.9
97	1.6	80	2	2.9
98	1.6	80	2	2.9
99	1.6	75	4	3.06
100	1.6	75	4	3.06
101	1.6	75	4	3.06
102	1.6	75	4	3.06
103	3.8	38	6	3.7
104	3.8	38	6	3.7
105	3.8	38	6	3.7
106	3.8	38	6	3.7
107	3.8	38	6	3.7
108	3.8	38	6	3.7
109	1	90	6	2.2
110	1	90	6	2.2
111	1	90	6	2.2
112	1	90	6	2.2
113	1	90	6	2.2
114	2.4	50	8	4
115	2.4	50	8	4
116	2.4	50	8	4
117	2.4	50	8	4
118	2.4	50	8	4
119	2.4	50	8	4
120	2.4	50	8	4
121	3	45	4	3.3
122	3	45	4	3.3
123	3	45	4	3.3

124	3	45	4	3.3
125	2.55	50	6	3.05
126	2.55	50	6	3.05
127	2.55	50	6	3.05
128	2.55	50	6	3.05
129	2.55	50	6	3.05
130	2.55	50	6	3.05
131	4	32	8	4.16
132	4	32	8	4.16
133	4	32	8	4.16
134	2.36	60	6	2.16
135	2.36	60	6	2.16
136	2.36	60	6	2.16
137	2.36	60	6	2.16

3. Code source:

Comme mentionné précédemment, nous avons écrit trois programmes « sélection aléatoire », « sélection en fonction de temps d'apprentissage », et notre approche « sélection aux multicritères _k-means », qui sont programmées comme indiqué ci-dessus :

3.1. Sélection aléatoire :

La figure illustre au niveau du fichier (serveur.cc) le programme choisis 89 clients aléatoire et envoyez-leur le modèle d'entrainement comme suit :

```
void Serveur::initialize()
{

    if (strcmp("serveur",getName())==0){

        cMessage *msg=new cMessage("aleatoire");
        createListClient();
        for (int i=0;i<89;i++)
                    {
                    cMessage *copy = msg->dup();
                    send(copy,"port$o",t[i]);
                    }
    }
```

```
void Serveur::createListClient(){

    for (int i=0;i<89;i++)
    {
        repeat:
        int n=uniform(0,136);
        for (int j=0;j<i;j++){
            if (n==t[j])
                goto repeat;
            }
        t[i]=n;
```

Figure 20: Fonction de sélection aléatoire.

3.2. sélection en fonction de temps d'apprentissage :

Au début, au niveau du fichier serveur.cc le serveur envoi une demande d'information à tout les clients comme illustre la figure 26:

```
void Serveur::initialize()
{

    if (strcmp("serveur",getName())==0){

        cMessage *msg=new cMessage("aleatoire");
       // createListClient();

        for (int i=0;i<137;i++)
                    {
                    cMessage *copy = msg->dup();
                    send(copy,"port$o",i);
                    }
```

Figure 21: La fonction qui renvoi une demande de ressource par le serveur.

Puis au niveau du fichier client.cc chaque client renvoi ses paramètres au serveur, qui est représenté dans le temps d'apprentissage principalement et « CPU, RAM, fréquence » juste pour comparais.

```
void Client::handleMessage(cMessage *msg)
{
    myMessage *msgTime=createMessage();
    send(msgTime,"port$o");
}
 myMessage *Client::createMessage(){

     double cpu=par("capaciteCPU");
     double memoire=par("tailleMemoire");
     double frequence=par("frequence");
     double temp=par("tempApprentissage");
     char msgName[30];
     sprintf(msgName,getName());
     myMessage *msg=new myMessage(msgName);
     msg->setTemp(temp);
     msg->setCPU(cpu);
     msg->setFrq(frequence);
     msg->setMemoire(memoire);

  return msg;
 }
```

Figure 22: La fonction qui renvoi les paramètres du client.

Ensuite, le serveur sélectionne les clients ayant le temps d'apprentissage inferieur ou égale à 90 s, pour participer au tour d'apprentissage.

```
void Serveur::handleMessage(cMessage *msg)
{
myMessage *msgNew = check_and_cast<myMessage *>(msg);
addToList(msgNew);

}
void Serveur::addToList(myMessage *msg){

    if(msg->getTemp()<=90.0){
        cl[i]=msg->getName();
        EV<<"le client "<<i<< " "<<cl[i]<<" a le temp : "
        <<msg->getTemp()<<", cpu: "<<msg->getCPU() <<
        ", memoire: "<<msg->getMemoire()<<", frequence:"
        <<msg->getFrq()<<"\n";
        i++;
    }
}
```

Figure 23: Le code source de la sélection selon le temps d'apprentissage.

3.3. sélection aux multicritères _k-means :

Dans cette méthode le serveur envoi une demande des paramètres à tous les clients, puis faire le clustering selon le type d'apprentissage non supervisé « k-means ».

```
void Serveur::handleMessage(cMessage *msg)
{
    myMessage *msgNew = check_and_cast<myMessage *>(msg);

    x=msgNew->getCPU();
    y=msgNew->getTemp();
    k=msgNew->getMemoire();
    z=msgNew->getFrq();

     point *p=new point(x,y,k,z,-1);
     list[conteur]=p;

       Kmeans();
       EV<<"cluster: "<<list[conteur]->getCluster()<<
       " temp: "<<y <<", cpu: "<<x<<", memoire: "<<k
       <<", frequence:"<<z<<"\n";
       conteur++;
}
```

Figure 24: Le serveur fait un appel de fonction à la fonction k-means.

Nous avons fixé le nombre de cluster à deux cluster, le premier est celui choisi par le LP Assistant et le deuxième a été choisi aléatoirement parmi les

mauvais clients, qui ont le centroid respectivement centroidMax, et centroidMin comme le montre la figure 30:

```
void Serveur::Kmeans(){
    // initialize centers
         point *centroidMax=new point(x1,y1,k1,z1,-1);
         point *centroidMin=new point(x2,y2,k2,z2,-1);
//calculate distance
    double dist1= list[conteur]->distance(centroidMax);
    double dist2= list[conteur]->distance(centroidMin);

         if(dist1<=dist2)
    {
              list[conteur]->setCluster(1);
    }
    else
    {
              list[conteur]->setCluster(2);
    }
}
```

Figure 25: Le code source du k-means en c++.

Le k-means dépend dans ses choix du calcul d'une distance calculée comme suit :

```
return (p->x - x) * (p->x - x) + (p->y - y) * (p->y - y)+ (p->k - k)
        * (p->k - k)+ (p->z - z) * (p->z - z);
```

Figure 26: Distance euclidienne.

1. Résultats :

Après avoir définir les valeurs de centroïde max à (CPU=2.35 GHz, temps=30 s, RAM=4 Go, fréquence= 0.16 GHz) et le centroïde min à (CPU=1 GHz, temps=90 s, RAM= 2 Go, fréquence=0.1 GHz), ainsi le nombre de clients nécessaires pour chaque tour à 89 clients, nous avons exécuté le programme et obtenu les résultats suivants :

Comparaison selon les capacités de CPU :

<Sélection aléatoire>

CPU(GHz)	Nombre de client
0-1	7
1-2	22
2-3	32
3-4	19
4-5	9
Total général	**89**

<Sélection par temps d'apprentissage>

CPU(GHz)	Nombre de client
1-2	26
2-3	31
3-4	20
4-5	12
Total général	**89**

<Sélection avec K-means>

CPU(GHz)	Nombre de client
2-3	45
3-4	29
4-5	15
Total général	**89**

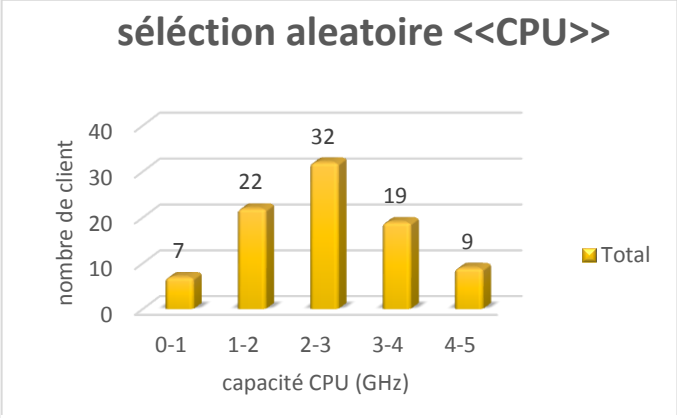

Figure 27: Les clients sélectionnés par la fonction aléatoire selon la capacité CPU(GHz).

Analyse :

- Nous remarquons que dans la sélection aléatoire le nombre de client ayant une capacité CPU entre 2 et3 GHz est le plus élevé où il atteint 32 clients, c'est un bon nombre pour une capacité moyenne.
- Le nombre de client ayant une capacité CPU entre 3 -4 GHz et 4-5 GHz sont respectivement 19 et 9 clients.
- Un nombre indésirable arrive à 29 clients ont été choisis ayant une capacité entre 0 et 2 GHz.

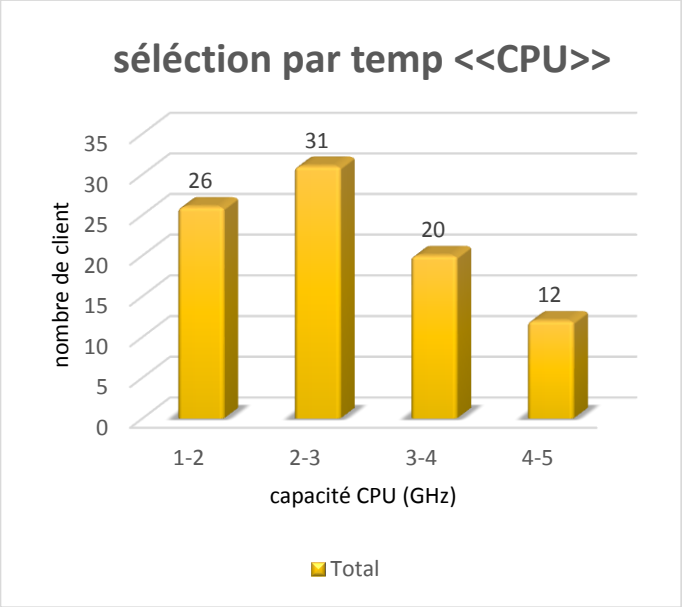

Figure 28: Les clients sélectionnés par la fonction du temps selon la capacité CPU(GHz).

Analyse :

- Nous notons que cette méthode a sélectionné 31 clients ayant une capacité entre 2 et 3 GHz, et une somme de 32 clients ayant une bonne capacité (entre 3 et 5 GHz).
- Nous notons aussi qu'il sélectionne un nombre important (26 clients) qui ont une capacité entre 1 et 2 GHz.

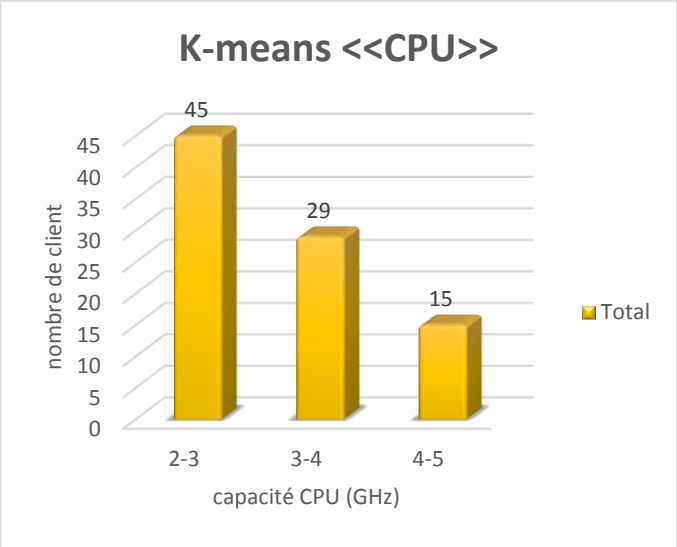

Figure 29: Les clients sélectionnés par K-means selon la capacité CPU(GHz).

Analyse :

- Nous notons un très bon nombre de client ayant une capacité CPU moyenne arrive à 45 clients, et un nombre très important des bons clients représenté par 44 clients ayant de 3 à 5 GHz de capacité CPU. Cette méthode ne sélectionne que des clients ayants une capacité de moyenne à élever, c'est le but que nous voulons atteindre.

Conclusion :

Nous remarquons clairement la supériorité de cette dernière méthode (K-means), où elle donne comme résultats 45 clients moyens et 44 clients excellents.

Comparaison selon le temps d'apprentissage

<Sélection aléatoire>

temps(s)	Nombre de client
30-39	21
40-49	10
50-59	14
60-69	13
70-79	3
80-89	10
90-99	7
110-120	11
Total général	**89**

<Sélection par temps d'apprentissage>

temps(s)	Nombre de client
30-39	28
40-49	7
50-59	11
60-69	14
70-79	8
80-90	21
Total général	**89**

<Sélection avec K-means>

temps(s)	Nombre de client
30-39	36
40-49	11
50-60	42
Total général	**89**

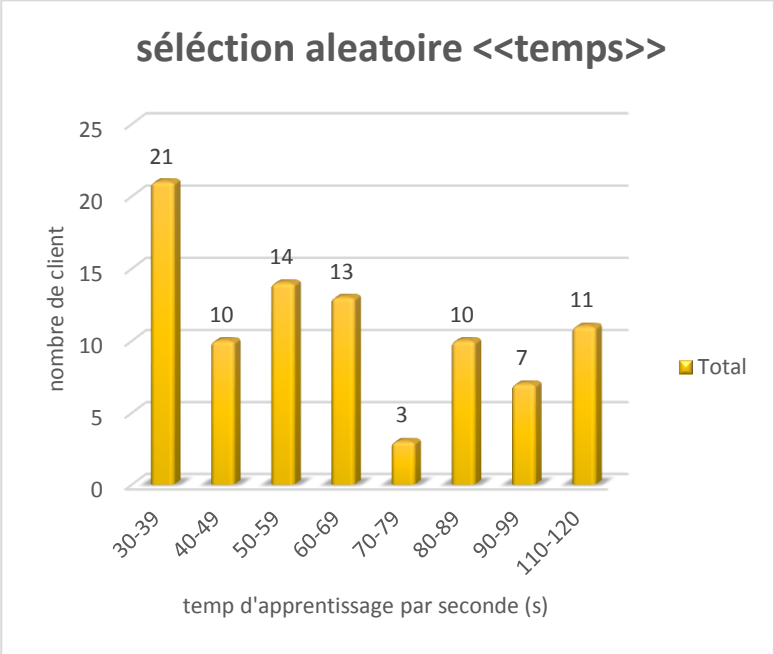

Figure 30: Les clients sélectionnés par la fonction aléatoire selon le temps d'apprentissage(s).

Analyse :

- Nous remarquons que cette méthode a sélectionné la moitié des clients de bonne qualité telle que 45 clients prennent entre 30 et 59 s pour faire l'apprentissage.
- 44 clients prennent entre 60 et 120 s ce qui consiste une longue durée provoquant ainsi l'élimination du tour.

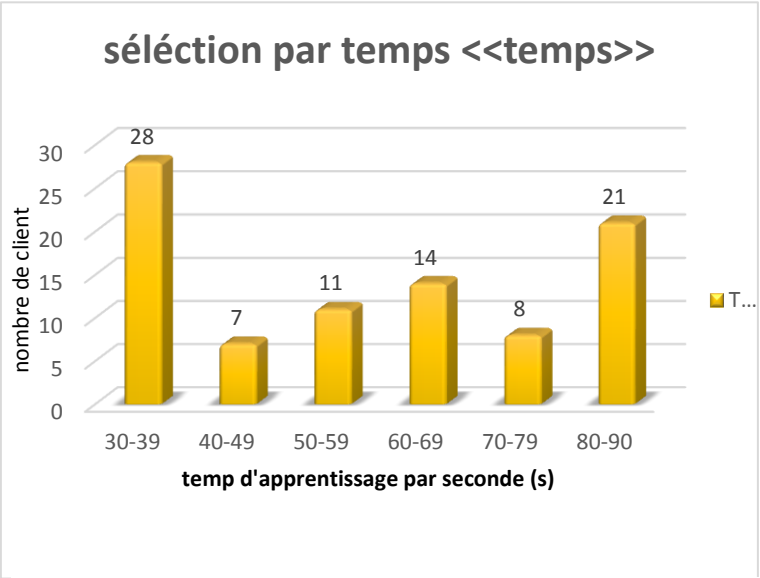

Figure 31: Les clients sélectionnés par la fonction du temps selon le temps d'apprentissage(s).

Analyse :

- nous nous apercevons que malgré les 46 clients ayant un temps d'apprentissage bas (30-59 s), les 43 restants, ayant le temps entre 60-90 s, peuvent nuire au tour.

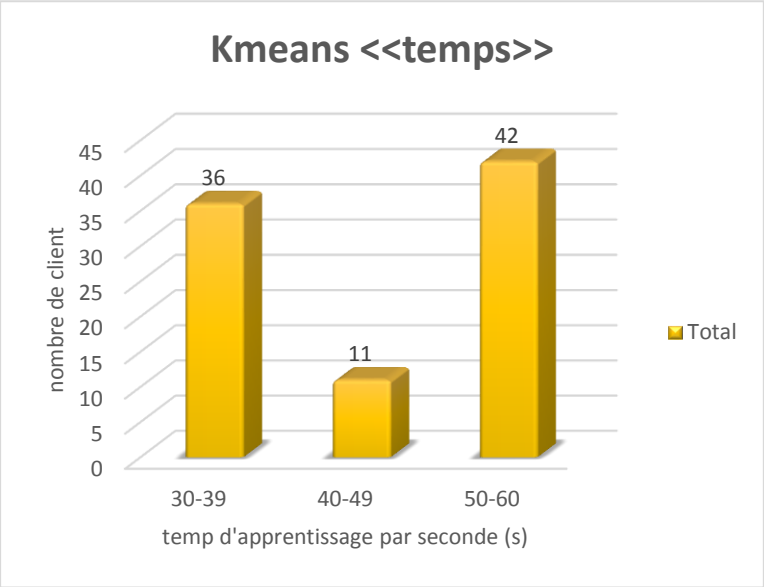

Figure 32: Les clients sélectionnés par K-means selon le temps d'apprentissage(s).

Analyse :

- nous remarquons que les clients possèdent un temps d'apprentissage favorable.

Conclusion :

Le K-means a des résultats plus avantageux que les autres méthodes.

Le K-means a des résultats plus avantageux que les autres méthodes. Comme dans notre cas d'étude, les appareils IOT, tels que les électro-ménagers ne sont pas aussi puissants qu'un serveur ou autres, pour cela l'algorithme du K-means sélectionne les appareils les plus puissant dans ce domaine ainsi que les moyens afin de maximiser le nombre de participants, et par conséquent le modèle global converge plus rapidement à la précision souhaitée.

Comparaison selon la taille de mémoire(RAM) :

<Sélection aléatoire>

taille mémoire(Go)	Nombre de client
0-2	4
2-4	10
4-6	20
6-8	23
8-10	23
16-18	3
32-34	4
62-64	2
Total général	**89**

<Sélection par temps d'apprentissage>

taille mémoire(Go)	Nombre de client
2-3	13
4-5	31
6-7	15
8-9	18
16-17	5
32-33	4
62-64	3
Total général	**89**

<Sélection avec K-means>

taille mémoire(Go)	Nombre de client
2-3	4
4-5	18
6-7	30
8-9	25
16-17	5
32-33	4
62-64	3
Total général	**89**

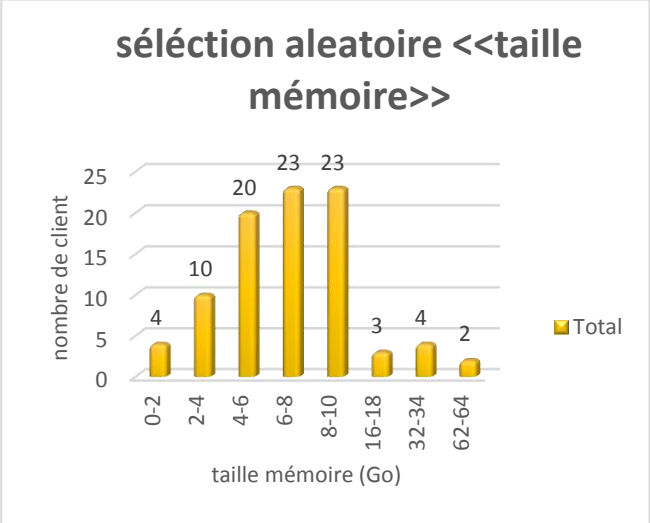

Figure 33: Les clients sélectionnés par la fonction aléatoire selon la taille de la mémoire RAM (Go).

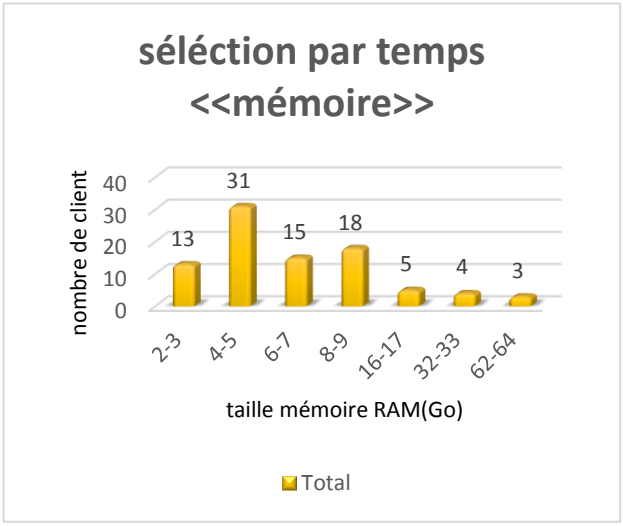

Figure 274: Les clients sélectionnés parla fonction du temps selon la taille de la mémoire RAM (Go).

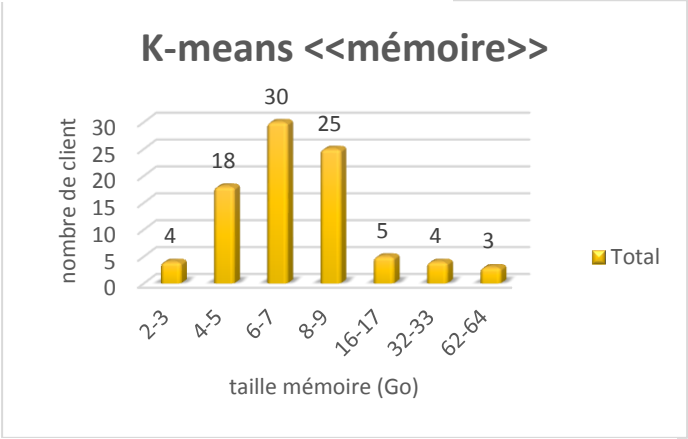

Figure 35: Les clients sélectionnés par K-means selon la taille de la mémoire RAM (Go).

Conclusion :

La majorité des clients sélectionné par le k-means possèdent une bonne capacité mémoire (RAM) comparais aux autres méthodes.

Comparaison selon la fréquence (GHz) :

<Sélection aléatoire>

fréquence(GHz)	Nombre de client
0-1	7
1-2	9
2-3	36
3-4	16
4-5	21
Total général	**89**

<Sélection par temps d'apprentissage>

fréquence (GHz)	Nombre de client
1-2	12
2-3	44
3-4	8
4-5	25
Total général	**89**

<Sélection avec K-means>

fréquence(GHz)	Nombre de client
1-2	4
2-3	31
3-4	19
4-5	35
Total général	**89**

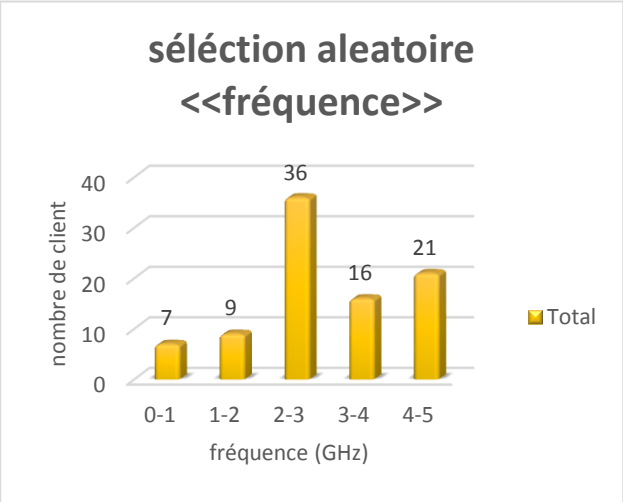

Figure 36: Les clients sélectionnés par la fonction du temps selon la fréquence (GHz).

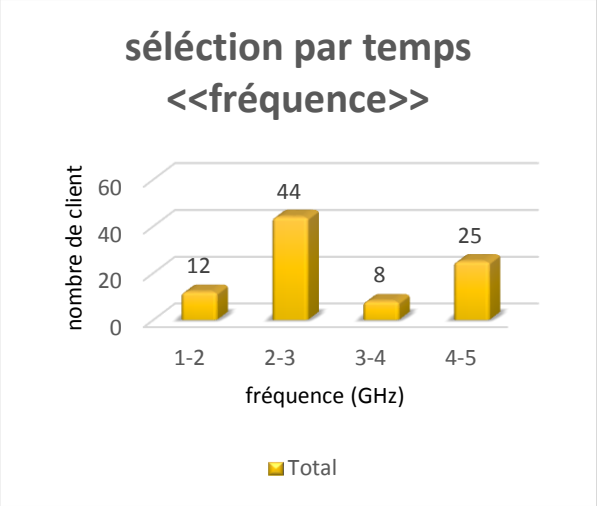

Figure 37: Les clients sélectionnés par la fonction aléatoire selon la fréquence (GHz).

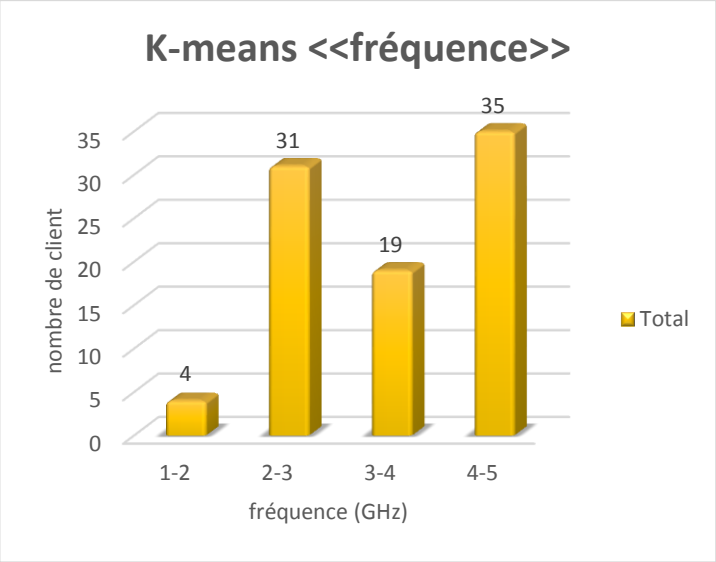

Figure 38: Les clients sélectionnés par K-means selon la fréquence (GHz).

Conclusion :

Le k-means a sélectionné des clients ayant une bonne fréquence avec 35 clients ayant entre 4-5 GHz, 31 clients ayant entre 2-3 GHz et 19 clients ayant entre 3-4 GHz.

Le K-means est remarquablement efficace en termes de sélection compare à la méthode de sélection aléatoire et la sélection par rapport au temps.

5.Conclusion :

En résumé, dans ce chapitre nous avions cité en premier lieu l'environnement de développement de notre approche, puis, nous avons présenté l'architecture de notre projet ainsi que la base de données utilisées pour les tests, nous avons également permis la visualisation de celui-ci grâce notamment aux captures d'écran, à la fin nous avons manifesté les résultats de ces tests.

Nous remarquons que notre méthode (k-means) a vraiment amélioré les résultats par rapport à la sélection aléatoire et la sélection en fonction de temps d'apprentissage, grâce au recours à plusieurs critères, et non à un seul. Notre approche a réussi à sélectionner que des clients nous considérons comme de bonne ou moyenne qualité, ce qui réduit la possibilité des « drops outs » et maximise en même temps le nombre de clients qualifiés qui participes à chaque tour.

CONCLUSION GENERALE

1. Conclusion :

Le terme “Internet des objets” fait référence à la connectivité et à la mise en réseau des objets et des systèmes du quotidien, de la société ou de l’entreprise. Il permet l’automatisation et l’imbrication de nombreux processus, que ce soit dans le domaine de la logistique, de la santé ou encore de la domotique. L’Internet des objets concerne une immense panoplie d’objets, du téléphone intelligent à la montre connectée en passant par les systèmes d’alarme au domicile ou encore les capteurs de vérification de l’air. Pour rendre les bâtiments intelligents entièrement automatiques et plus précis, les scientifiques ont développée l’apprentissage fédéré afin d’entrainer des modèles dans les IoT. Le FL consiste à sélectionner un nombre d'appareils via un serveur, ces appareils entrainent à leur tour le modèle en utilisant leurs données locales. Cette partie, qui consiste à sélectionner les clients, est devenue un défi du choix des clients optimaux.

Dans notre étude nous avons proposé une nouvelle approche surnommée sélection aux multi critères, cette technique est basée sur l’algorithme non supervisé k-means. Le but est de diminuer le nombre de tour d’apprentissage nécessaire et de maximiser les clients capables de réaliser l’entrainement tout en réduisant les drops outs .la méthode se base sur 4 critères : la capacité CPU, le temps d’apprentissage, la fréquence de la bande passante et la taille mémoire(RAM). Initialement, elle désigne deux centres : un optimale et un autre mauvais, ensuite lance le k-means qui nous donne en résultats deux clusters : un cluster comportant les clients ayant les compétences moyennes à élevées, l’autre comportant les clients les moins compétents pour notre cas.

Pour prouver l’efficacité de notre méthode, nous l’avons simulé avec deux autres méthodes (sélection en fonction du temps/sélection aléatoire), puis comparé entre les résultats des trois méthodes.

Les résultats de notre expérimentation montrent que notre approche a prouvé son efficacité ainsi que l’amélioration de qualité des clients sélectionnés.

2. Perspective :

Cependant nous pensons qu’un certain nombre de points restent à explorer, tel que : l’ajout d’autres critères (connectivite, énergie…), trouver une méthode qui calcule les coefficients, rendre le programme linéaire personnalisable.

Du point de vue pratique, une perspective très évidente serait d’automatisé le processus en entier pour avoir un outil de sélection automatique.

Références :

[1] Adrian PETRICEVIC, Vincent TALLOIR.(2019). L'intelligence artificielle : une solution à la prise de décision quotidienne ?

[2] Dorsemaine, B., Gaulier, J. P., Wary, J. P., Kheir, N., & Urien, P. (2015, September). Internet of things: a definition & taxonomy. In 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies (pp. 72-77). IEEE.

[3] Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. The internet society (ISOC), 80, 1-50.

[4] Ashton, K. (2009). That 'internet of things' thing. RFID journal, 22(7), 97-114.

[5] KHIRI Tarek.(2019). Développement d'un système d'évacuation guidée en utilisant la réalité virtuelle.

[6] Maaza Hanane, Benmenni Amina.(2021). Conception d'une maison intelligente avec les réseaux M2M/IoT.

[7] ADDOU Asmaa, BAHOUS Nawel.(2020). Réalisation d'une maison intelligente à base d'Arduino.

[8] YAHI Amina, KOURI Loubna.(2018). Contrôle et suivi d'une maison intelligente via Internet.

[9] AID Lahcene.(2016). Modélisation et simulation du confort dans un bâtiment intelligent par le formalisme DEVS.

[10] Marine BELLAS, Maxime QUEMIN.(2017). Smart building : construction de bâtiment intelligent, apport de l'électronique imprimée.

[11] Mr. BRAHIM Aimen, Mr. NEBIH Akram. (2020). Classification des images satellitaires pour l'aide à la gestion des catastrophes naturelles en utilisant l'apprentissage profond.

[12] BENSIAH Oussama Akram. (2020).Proposition d'une nouvelle approche basée Deep Learning pour la prédiction du cancer du sain.

[13] datascience.eu/fr/apprentissage-automatique/apprentissage-federe.

[14] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *13*(3), 1-207.

[15] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR.

[16] Li, Y., Chang, T. H., & Chi, C. Y. (2020, September). Secure federated averaging algorithm with differential privacy. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1-6). IEEE.

[17] AbdulRahman, S., Tout, H., Mourad, A., & Talhi, C. (2020). FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IEEE Internet of Things Journal*, *8*(6), 4723-4735.

[18] Nishio, T., & Yonetani, R. (2019, May). Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)* (pp. 1-7). IEEE.

[19] Mohammed, I., Tabatabai, S., Al-Fuqaha, A., El Bouanani, F., Qadir, J., Qolomany, B., & Guizani, M. (2020). Budgeted online selection of candidate IoT clients to participate in federated learning. *IEEE Internet of Things Journal*, *8*(7), 5938-5952.

[20] www-igm.univ-mlv.fr/~badis/ESIEE/TP1-MDR.htm.