Année Universitaire 2023 – 2024
Unité de Recherche : LIASD - LIPAH
Équipe(s) : Projet PHC-Utique
Encadrant(s) : Hella Kaffel & Akram Hakiri & Nedra Mallouli

Modèles d'Apprentissage Profond Fédérée Explicable sur Jumeaux Numériques pour l'Agriculture Intelligente

Karim Houidi

Janvier 2024

Résumé

Le stage de thèse s'intègre dans un projet PHC-Utique qui a été réalisé en collaboration entre l'université de Paris 8 et l'université Tunis El Manar. Le titre du sujet de thèse est « modèles d'apprentissage profond fédérés explicables sur jumeaux numériques pour l'agriculture intelligente ».Ce résumé met en lumière le travail réalisé dans le cadre du stage de thèse, en mettant l'accent sur l'état de l'art qui a posé les bases théoriques nécessaires à la poursuite des travaux de recherche.

Mots clés : Jumeau numérique, apprentissage fédéré, IoT, Intelligence artificielle

Sommaire

1 Introduction

Le stage de recherche de la thèse permet aux étudiants d'apprendre le métier de chercheur, en participant au travail de recherche d'un laboratoire. Le travail d'un chercheur ressemble beaucoup à celui d'un journaliste : il faut d'abord avoir l'idée d'un sujet d'enquête, ensuite, il faut que l'enquête fournisse des informations et, enfin, il faut la publier.
Le cadre du projet PHC-Utique est défini par un ensemble de paramètres et de directives qui orientent sa mise en œuvre. Ce projet vise à promouvoir la coopération scientifique et technologique entre la France et la Tunisie dans les domaines de la recherche et de l'innovation. Dans ce contexte, le cadre du projet PHC-Utique comprend des objectifs clairement définis, des partenariats solides entre les institutions des deux pays, l'université Tunis El Manar, la faculté des sciences de Tunis (Laboratoire en Informatique en Programmation Algorithmique et Heuristique, LIPAH) et l'université Paris 8, l'IUT de Montreuil (Laboratoire d'Intelligence Artificielle et Sémantique des Données LIASD) ainsi que des mécanismes de suivi et d'évaluation rigoureux. En outre, il s'inscrit dans une démarche de développement durable et de valorisation des compétences locales.

2 État de l'art

2.1 Introduction

L'agriculture, principal secteur de consommation d'eau, utilise la plupart des réserves l'eau disponible, représentant près de 40,03 pourcent des ressources disponibles. L'agriculture pluviale est contrainte par l'aridité du climat, l'irrégularité des précipitations et la dégradation de la qualité des sols due à l'érosion. La gestion efficace de l'eau potable et d'irrigation dans les régions arides et semi-arides pose aujourd'hui un problème majeur au sommet des priorités du développement mondial.
Dans cette thèse de doctorat, un système à base des jumeaux numériques (Digital Twin) dont les principaux composants sont :

- Collecte efficace de données agricoles en temps réel à travers des réseaux de capteurs agricoles à faible consommation d'énergie.
- Analyse et traitement des données collectées dans des endroits proches des champs et des parcelles d'étude.
- Construction des modèles explicatifs qui intègrent le contexte et l'environnement dans lesquels ils opèrent.
- Conception d'un système de recommandation collaboratif pour fournir des suggestions aux agriculteurs en fonction de données multi-sources telles que les données climatiques, les données satellitaires, l'état des nappes phréatiques, les précipitations, de la température, de la superficie des terres, du rendement des cultures passées et d'autres paramètres, réduisant ainsi l'effort et le temps requis pour les différents processus agricoles. Ce système de recommandation permettra de sensibiliser les plus jeunes aux métiers d'agriculteur afin de valoriser l'agriculture, en proposant de vivre une expérience d'immersion sur des sites et des lieux réels.

Cette thèse porte également sur la conception d'algorithmes d'analyse de données massives distribuées agricoles et de techniques d'imagerie pour optimiser l'irrigation et réduire l'application de pesticides et d'herbicides, ainsi que le développement du jumeau numérique agricole et du système de recommandation agricole intelligent pour former les agriculteurs et sensibiliser les plus jeunes aux métiers de la ferme

2.2 Les jumeaux numériques

Le jumeau numérique (DT) est une représentation numérique d'un actif physique qui peut être utilisée pour décrire ses propriétés, son état et son comportement par la modélisation, l'analyse et la simulation. Les jumeaux numériques peuvent aider les machines-outils à effectuer leurs tâches de surveillance et de

dépannage de manière autonome du contexte de la fabrication intelligente. Pour cela, un type spécial de jumeau appelé jumeau basé sur le signal du capteur doit être construits et adaptés dans les systèmes cyber-physiques. Le jumeau doit apprendre automatiquement la connaissance des ensembles de données de signaux de capteur historiques, interagir de manière transparente avec les signaux de capteur en temps réel, gérer les ensembles de données sémantiquement annotés stockés dans les nuages, et s'adapter au délai de transmission des données. Le développement de tels jumeaux n'a pas encore été étudié en détail.

2.3 Conception de jumeaux numériques

Le concept DT a été inventé par M.Grieves dans un livre blanc (S'afflige, 2014) en tant qu'unification des actifs virtuels et physiques dans le cyle de vie du produit. Résultat : outils logiciels commerciaux pour développer DT (Predix et Simcenter). Autre concept des jumeaux numériques remonte aux années 1960, lorsque la NASA a lancé l'idée du jumelage dans le cadre de son programme Apollo visant à créer des doubles physiques sur Terre qui correspondent à leurs systèmes dans l'espace. L'idée leur a permis de simuler divers scénarios, de tester différents cas et conditions et d'évaluer le comportement et les performances de leurs systèmes. Il a pris de l'élan lorsque le jumeau est venu à la rescousse après que des problèmes techniques dans la mission Apollo 13 ont été résolus par des ingénieurs sur terre en testant des solutions possibles sur le jumeau au sol. Plus tard, ce n'est qu'au début des années 2000 que Michael Grieves a introduit le concept de jumeaux numériques pour l'industrie manufacturière en créant des répliques virtuelles d'usines pour surveiller leurs processus, prévoir les défaillances et augmenter leur productivité. Le concept a gagné en attention et en ampleur après avoir été classé parmi les 10 principales tendances technologiques stratégiques en 2017 par Gartner, et adopté par de nombreux géants de l'industrie comme Siemens et General Electric.

- le jumeau numérique peut être conçu de deux manières principales. Une possibilité consiste à créer un modèle système de l'objet physique. L'autre possibilité consiste à créer une structure de données qui organise et relie les données du capteur et d'autres informations.
- Digital Twin ontologies : DT qui utilise des ontologies pour permettre la co-évolution avec le CES (systèmes d'ingénierie complexes) en assimilant les données en termes de variété, de vitesse et de volume tout au long du cycle de vie des actifs.
  - Collecte et stockage de cycle de vie du produit, utilisant technologie de l'IdO, possible avec des plates-formes PLM ou ERP
  - Gestion des données : Approches décentralisées(-plus redondants,+plus sûres, -beaucoup d'efforts pour diffuser les changements), Approches centralisées (+meilleures pour gérer le changement de données,+moins redondants, +utilisent un référentiel central pour gérer et fournir de données à chaque logiciel à la demande, Approches linguistiques partagées : conservent les avantages centralisateurs
- Architectures abstraites qui soutiennent la définition du système sous-jacent d'un DT et de ses éléments structurels de base. L'une des architectures les plus connues est RAMI, qui fournit des packages structurés et des clusters, y compris des couches DT, et dont l'élément central est l'Asset Administration Shell (AAS). L'AAS est la représentation numérique d'un actif, qui peut consister en sous-modèles, avec leurs propriétés, les opérations qu'il peut effectuer et les événements qui lui sont associés.

Un exemple de plateforme qui prend en charge la création de DT est uDiT (Universal Digital Twin Platform). Il fournit un middleware de communication basé sur OMG DDS, une interface de moteur de temps d'exécution de middleware et DT, des fonctions de co-simulation basées sur l'interface de maquette fonctionnelle (FMI), et des passerelles pour la conversion de médias et de protocoles. Apache Kafka, qui prend en charge la communication de données distribuée en temps réel. Sur la base de ce courtier, une solution de micro-service a été mise en œuvre pour soutenir le développement de DT.
Solutions commerciales : Siemens propose MindSphere, une plateforme capable de gérer un nombre pertinent de flux de données et de connecter des appareils et des machines pour créer des DT. General Electric propose Predix, une plate-forme de soutien à la création de DT pour l'analyse et la surveillance,

capable de collecter des données issues de nombreux processus de fabrication ou industriels. IBM propose la plateforme Watson IoT capable de gérer les données en temps réel, qui offre des fonctionnalités supplémentaires telles que l'analyse des données et les services basés sur le cloud.
Solutions open source telles que Ditto et le RAMI AAS permettent de pallier le manque de ressources limitées. La première est une plateforme axée sur la fourniture d'une solution fiable pour le développement de DT. Ditto s'appuie sur le concept de Thing, qui sont décrits par Features and related information model. Comme AAS utilise le concept de sous-modèles, Ditto est basé sur des blocs fonctionnels, qui permettent d'organiser des propriétés, des actions et des événements. FIWARE est une autre possibilité open-source bien connue, ou les DT sont des entités numériques représentant un actif physique réel.
La plateforme Clawdite a été conçue pour répondre au besoin récurrent d'une solution capable de prendre en charge la création de DT dans différentes applications de fabrication.
Digital Twin Augmenté : un système complexe qui interagit non seulement avec son entité réelle, mais aussi avec son environnement et d'autres jumeaux numériques. Le système de jumeaux numériques augmentés comprend la contrepartie numérique et son environnement, la relation avec d'autres jumeaux numériques, l'entité physique et son environnement, la relation avec d'autres entités physiques. Ils communiquent entre eux, changent simultanément, interagissent et se touchent mutuellement.
Juméau Numérique Hmain : HDT est basé sur le modèle Augmenté Digital Twin, compose de deux parties, l'entité physique, la contrepartie virtuelle et la communication bidirectionnelle entre elles, des entourages et d'autres entités (ici d'autres personnes réelles) sont ajoutés à l'espace physique, et des environnements virtuels et d'autres jumeaux numériques sont ajoutés au cyberespace.
Il n'existe pas de définition unique du jumeau numérique, et de nombreux auteurs ont élaboré leurs propres définitions. Par exemple, un jumeau numérique doit ^etre défini comme une représentation numérique d'un objet du monde réel en se concentrant sur l'objet lui-même, ou comme une simulation intégrée, multiphysique, multi-échelle et probabiliste d'un système tel qu'il est construit, rendue possible par Digital Thread, qui utilise les meilleurs modèles disponibles, les informations de capteur et les données d'entrée pour refléter et prédire les activités/performances pendant la vie de son jumeau physique correspondant.
Le modèle conceptuel de Digital Twin comporte essentiellement trois parties principales :

1. Produit physique dans l'espace physique ;
2. la contrepartie virtuelle du produit physique dans le cyberespace ;
3. Interface d'interaction de données et d'informations entre l'espace physique et le cyberespace.

2.4 Apprentissages automatiques

L'objectif principal est d'acquérir, fusionner et analyser des données dans le domaine de l'agriculture. Cela inclut la fusion en quasi-temps réel de données spatio-temporelles provenant de sources diverses et hétérogènes. De plus, l'analyse et la prédiction de séries temporelles multivariées et multi-temporelles ainsi que l'explicabilité et la traçabilité des algorithmes élaborés sont des objectifs clés.

En ce qui concerne le jumeau numérique agricole, l'objectif est de mettre en place une architecture distribuée, autonome et collaborative pour le jumeau numérique. Il est également important de penser aux interfaces du jumeau numérique afin de faciliter et de répondre aux besoins des différents acteurs impliqués dans le domaine de l'agriculture.

2.4.1 Définition

Le machine learning, ou apprentissage automatique, est une branche de l'intelligence artificielle qui consiste à concevoir des algorithmes capables d'apprendre à partir de données et de s'améliorer de façon autonome. Ces algorithmes sont utilisés dans de nombreux domaines tels que la reconnaissance d'images, la prédiction de données, la recommandation de produits, et bien d'autres applications. En somme, le machine learning permet aux machines d'acquérir des connaissances et des compétences sans être explicitement programmées pour chaque tâche. ML est un champ d'étude de l'intelligence

artificielle qui vise à donner aux machines la capacité d'apprendre à partir de données via des modèles mathématique, une branche de l'intelligence artificielle (IA) qui se concentre sur l'utilisation de réseaux de neurones artificiels profonds pour résoudre des problèmes complexes.

Le deep learning, ou apprentissage profond, est une branche de l'intelligence artificielle qui se concentre sur l'entraînement de réseaux de neurones artificiels pour apprendre et effectuer des tâches complexes. Cette technique est utilisée dans de nombreux domaines tels que la reconnaissance d'images, la compréhension du langage naturel, la traduction automatique, et bien d'autres applications. L'apprentissage par renforcement permet d'obtenir des performances remarquables dans la résolution de problèmes qui étaient auparavant difficiles, voire impossibles à traiter pour les ordinateurs.

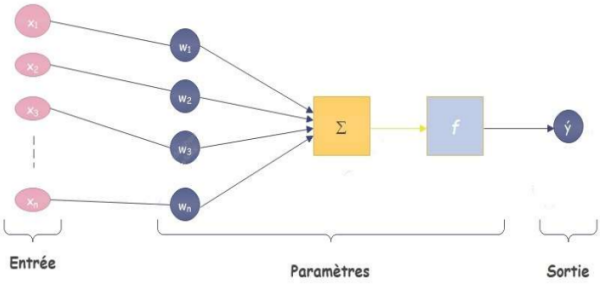

FIGURE 1 – Réseau de Neurone

2.4.2 Machine learning & Deep learning

La différence entre le machine learning et l'apprentissage par renforcement réside dans leur domaine d'application, le volume des données, le temps d'apprentissage et la puissance de calcul. Le machine learning est souvent utilisé dans des domaines tels que la reconnaissance de formes, la classification et la prédiction, avec des volumes de données relativement petits. L'apprentissage par renforcement, en revanche, est plus adapté pour des problèmes complexes tels que la reconnaissance d'images et de voix, avec des volumes de données massifs. En ce qui concerne le temps d'apprentissage, l'apprentissage par renforcement nécessite généralement plus de temps en raison de la complexité des modèles. Enfin, l'apprentissage par renforcement nécessite également une puissance de calcul plus importante en raison du grand nombre de paramètres à optimiser dans les réseaux de neurones profonds.

	Machine Learning	Deep Learning
Domaine d'application,	Classification, Régression, Clustering	NLP, Computer vision
Volume de données traitées	Milliers d'observations	Big Data
Temp d'apprentissage	Moins long	Plus longs
Puissance de calcul	Moins gourmand en ressources	GPU, TPU

TABLE 1 – ML & DL

2.5 Définitions de FL

L'apprentissage fédéré, également appelé apprentissage collaboratif, est une méthode d'apprentissage automatique qui permet de former des modèles de machine learning sur des données distribuées sur plusieurs appareils ou serveurs, sans avoir à les centraliser. Cela permet de préserver la confidentialité des données tout en bénéficiant des avantages de l'apprentissage automatique.

Références	Définitions
[1]	L'apprentissage fédéré (Federated Learning) propose d'avoir un ensemble de périphériques extrêmes pour effectuer des tâches d'apprenant localement et de ne communiquer qu'un modèle mis à jour à un serveur de coordination.Un serveur apprend un modèle global partagé en regroupant des modèles formés localement à partir d'un nombre potentiellement très important de clients. Les clients ont généralement des données déséquilibrées et non-i.i.d. (indépendant et distribué à l'identique) ainsi que des capacités de transfert de données limitées.
[2]	Federated Learning implique l'apprentissage collaboratif des modèles DNN (Deep Neural Network) sur les terminaux. Il y a, en général, deux étapes dans le processus de FL, l'apprentissage des modèles locaux sur les périphériques finaux et l'agrégation globale des paramètres mis à jour dans le serveur FL. FL permet aux utilisateurs de former en collaboration un modèle partagé tout en conservant des données personnelles sur leurs appareils, soulageant ainsi leurs préoccupations en matière de confidentialité.
[3]	Federated Learning, c'est une approche décentralisée qui plaide pour une alternative qui laisse les données d'entraînement distribuées sur les appareils mobiles, et apprend un modèle partagé en agrégeant les mises à jour informatisées localement.

TABLE 2 – Définitions FL

2.5.1 Les Algorithmes de FL

Federated Averaging (FedAvg)
Motivations [1] [4] : Motivations

- Un serveur apprend un modèle global partagé en regroupant des modèles formés localement à partir d'un nombre éventuellement très important de clients.
- Les clients ont généralement des relations déséquilibrées et non i.i.d (indépendant et distribué à l'identique).
- Capacités limitées de transfert de données.

2.5.2 Problèmes

Problème : Chercher un minimum w qui minimise la perte moyenne sur les n exemples d'entraînement. Dans un contexte big data : nombre d'exemples est trop important pour être stocké sur un seul ordinateur.

- Repartir le calcul sur plusieurs ordinateurs
- Le nombre d'exemples de formation détenus par le client k; $n_k = |P_k|$

$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w)$ avec $F_k(w) := \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$
K : client sur lequel les données et les calculs sont distribués
P_k : chaque client détient une partie P_k tous les exemples de formation et calcule $F(w)$: la perte moyenne du client k.
Problème d'optimisation : $\min_{w \in \mathbb{R}^d} f(w)$, $f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$
w : vecteur contient d paramètre de modèle Pour l'apprentissage supervisé $f_i(w)$, une fonction de perte (loss function) ; $f_i(w) = l(x_i; y_i; w)$.
(x_i, y_i) : l'un de n exemple de données étiquetés.

[1], [4], [5] : FedAvg orchestre la formation via un serveur central qui héberge le modèle global partagé

w_t ou t est le cycle de communication. L'optimisation effectuée localement sur les clients (utilisant par exemple, Stochastic Gradient Decent)SGD. FedAvg a cinq hyperparamètres :

1. la fraction de client C à sélectionner pour la formation
2. la taille du mini-lot local B
3. le nombre d'époques locales E
4. un taux d'apprentissage α (learning rate)
5. décroissance du taux d'apprentissage λ

2.5.3 Federated Stochastic Variance Reduced Gradient FSVRG

L'idée derrière FSVRG [1] est d'effectuer un calcul de coût d'apprentissage de gradient complet de manière centralisée,suivi de nombreuses mises à jour stochastiques distribuées sur chaque client.Le FSVRG standard n'a qu'un seul hyperparamètre : h. Le client k a une taille locale des étapes h_k qui est inversement proportionnelle à n_k, $h_k = \frac{h}{n_k}$

Algorithm 1: Federated Stochastic Variance Reduced Gradient FSVRG

Data: Initialisation : w_0

$h \leftarrow$ stepsize (Taille de pas)

$\{P_k\}_{k=1}^{K}$ = partition de données

for $t \leftarrow 0$ **to** ... **do**

  $\nabla f(w_t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w_t)$;

  for *Tous les clients k en parallèle* **do**

    initialisation : $w_{t+1}^k \leftarrow w_t$ et $h_k = \frac{h}{n_k}$;

    $\{i_s\}_{s=1}^{n_k}$ permutation de P_k ;

    for $s \leftarrow 1$ **to** n_k **do**

      $\Theta \leftarrow \nabla f_{i_s}(w_{t+1}^k) - \nabla f_{i_s}(w_t) + \nabla f_{i_s}(w_t)$;

      $w_{t+1}^k \leftarrow w_{t+1}^k - h_k \Theta$;

  $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$;

Result: w_{t+1}

2.5.4 Proposed Framework : FedProx

Motivations Les différents appareils dans les réseaux fédérés ont souvent des contraintes de ressources différentes en termes de matériel informatique, de connexions réseau et de niveaux de batterie. Par conséquent, il est irréaliste de forcer chaque appareil à effectuer une quantité uniforme de travail (c'est-à-dire exécuter le même nombre d'époques locales,E). **Principe** FedProx [6], est similaire à FedAvg dans le fait qu'un sous-ensemble de périphériques est sélectionné à chaque tour, des mises à jour locales sont effectuées, et ces dernières sont alors moyennées pour former une mise à jour globale. Toutefois, FedProx effectue les modifications simples mais critiques suivantes, qui donnent lieu à des améliorations empiriques importantes et nous permettent également de fournir des garanties de convergence. Dans FedProx :

— Généralisons FedAvg en permettant à des volumes de travail variables d'être effectués localement sur l'ensemble des appareils en fonction de leurs sources de ressources disponibles,

— Agrégant les solutions partielles envoyées par les retardataires(les appareils qui n'appartient pas ou sous ensemble sélectionné d'elle départ pour chaque tour).

En d'autres termes, au lieu d'assumer un γ uniforme pour tous les appareils tout au long du processus d'apprentissage, FedProx accueille implicitement des γ variables pour différents appareils et à différentes itérations. **Définition de γ-inexact solution** Pour une fonction $h(w, w_0) = F(w) + \frac{\mu}{2}\|w - w_0\|^2$ et $\gamma \in [0, 1]$, avec w_* est un γ-inexact solution de $\min_w h(w, w_0)$ si $\|\nabla h(w^*, w_0)\| \leq \gamma\|\nabla h(w_0, w_0)\|$ ou $\nabla h(w, w_0) = \nabla F(w) + \mu(w - w_0)$.
Définition de γ_k^t-inexact solution γ_k^t-inexact solution, pour l'appareil k à l'itération t, pour une fonction $h_k(w, w_t) = F_k(w) + \frac{\mu}{2}\|w - w_t\|^2$ et $\gamma \in [0, 1]$, avec w_* est un γ_k^t-inexact solution de $\min_w h_k(w, w_t)$ si $\|\nabla h_k(w^*, w_t)\| \leq \gamma_k^t\|\nabla h_k(w_t, w_t)\|$ ou $\nabla h_k(w, w_t) = \nabla F_k(w) + \mu(w - w_t)$.

Algorithm 2: FedProx (Proposed Framework)

Data: Paramètres : $K, T, \mu, \gamma, w_0, N, p_k, k = 1, \ldots, N$

for $t = 0$ **to** $T - 1$ **do**

Server sélectionne un sous-ensemble S_t de K appareils de manière aléatoire (chaque appareil k est choisi avec une probabilité p_k);

Le serveur envoie w_t à tous les appareils choisis;

Chaque appareil choisi $k \in S_t$ trouve un w_{t+1}^k qui est un minimiseur inexact de γ_t^k tel que :

$$w_{t+1}^k \approx \arg\min_w h_k(w; w_t) = F_k(w) + \frac{\mu}{2}\|w - w_t\|^2$$

Chaque appareil $k \in S_t$ envoie w_{t+1}^k de nouveau au serveur;

Le serveur agrège les w comme $w_{t+1} = \frac{1}{K}\sum_{k \in S_t} w_{t+1}^k$;

Result: w_{t+1}

2.5.5 Comparaissions entre les algorithmes de FL

Voici un tableau qui représente les algorithmes les plus utilisés pour l'apprentissage fédéré :

Référence	Algorithme	Description
[1]	FedAvg (Federated Averaging)	Moyenne des mises à jour des modèles locaux.$w_{t+1} = \sum_{k \in S_t} \frac{n_k}{n_\sigma} w_{t+1}^k$, où $n_\sigma = \sum_{k \in S_t} n_k$
[6]	FedProx (Federated Proximal)	Régularisation proximale γ_k^t-inexact solution pour encourager la similitude entre les modèles locaux et globaux.
[7]	FedOpt (Federated Optimization)	FedOpt vise à résoudre les défis liés à la distribution inégale des données, aux variations de la qualité des connexions et à d'autres problèmes spécifiques à l'apprentissage fédéré. Utilise des techniques d'optimisation avancées. CLIENTOPT et SERVEROPT sont des optimisateurs basés sur des gradients avec des taux d'apprentissage α_l et α respectivement. CLIENTOPT vise à minimiser en fonction des données locales de chaque client tandis que SERVEROPT optimise dans une perspective globale. FedOpt permet naturellement l'utilisation d'optimisateurs adaptatifs (par exemple, ADAM, YOGI, etc.), ainsi que de techniques telles que le dynamique côté serveur (FEDAVGM)
[8]	FedSGD (Federated SGD)	Utilise un algorithme de descente de gradient stochastique pour optimiser les modèles locaux.
[9]	q-FedAvg (Quantized Federated Averaging)	Version quantifiée de FedAvg, où les mises à jour du modèle sont quantifiées avant l'agrégation, q est un paramètre qui ajuste la quantité d'équité que nous voulons imposer.$\min_w f_q(w) = \frac{1}{m} \sum_{k=1}^m \frac{p_k}{q+1} F_k^{q+1}(w)$.
[10]	FedPer (Federated Personalization)	L'objectif principal de FedPer est de permettre aux modèles d'apprentissage d'être personnalisés pour chaque utilisateur tout en conservant la nature décentralisée de l'apprentissage fédéré. Chaque appareil ou client peut personnaliser son modèle en fonction de ses propres données locales. Intègre des mécanismes de personnalisation pour s'adapter aux préférences individuelles des appareils.

TABLE 3 – Algorithmes de Federated Learning

La comparaison entre les algorithmes de Federated Learning se fait au niveau de calcule des fonction de précision (Accuracy), qui mesure la proportion de prédictions correctes par rapport au nombre total d'échantillons, la perte (Loss), la fonction de perte utilisée pendant l'apprentissage peut être évaluée pour mesurer la qualité des prédictions. Pour être la comparaison plus dure et correcte en peut ajouter le temps de convergence, c'est le nombre d'itérations nécessaires pour atteindre une certaine précision ou pour que l'algorithme converge, en plus la communication entre le serveur et les clients, c'est le coût de la communication entre le serveur central et les clients peut être évalué, par exemple, en mesurant la quantité de données échangées ou la fréquence des communications.Pour faire tout les comparaisons nécessaire il faut bien implémenté les différent algorithme et le test sur des donnée réel comme MINIST ou CIFAR-10.

Caractéristique	FedAvg	FedSGD	FSVRC
Optimisation	Minimiser la pert moyenne sur n exemple, utilise une approche basée sur la moyenne des mises à jour des modèles des clients. $\min_{w \in \mathbb{R}^d} f(w), f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$	Utilise une approche de descente de gradient stochastique fédérée.	Calcule de coût d'apprentissage de gradient complet de manière centralisé, suivi de nombreuses mis à jours stochastiques distribuées sur chaque client.
Communication	Nécessite des communications entre les clients et le serveur central pour partager les mises à jour de modèle.	Implique également des communications entre les clients et le serveur central.	Communications avec les clients pour la mis à jours stochastique distribuées de modèle.
Agrégation	Utilise une moyenne pondérée des mises à jour des clients pour mettre à jour le modèle global.	Agrège généralement les mises à jour locales en utilisant des méthodes de moyenne ou d'autres mécanismes d'agrégation.	Agrégation avec réduction de gradient.
Complexité de l'algorithme	Relativement simple et facile à comprendre, en particulier grâce à son approche de moyenne pondérée.	Peut être plus complexe à mettre en œuvre et à ajuster en raison de la nature stochastique de la descente de gradient.	Introduit de l'hyperparamètre h, ajoutant une complexité.

TABLE 4 – Comparaison entre FedAvg, FedSGD, et FSVRC

2.5.6 Les outils de FL

Les outils de federated learning sont des technologies qui permettent de former des modèles d'apprentissage machine en utilisant des données distribuées sur plusieurs appareils ou serveurs, sans avoir à centraliser les données sur un seul emplacement. Ces outils incluent des frameworks de machine learning comme TensorFlow Federated, PySyft et Flower, qui facilitent la mise en œuvre du federated learning. En utilisant ces outils, les organisations peuvent entraîner des modèles de manière collaborative tout en préservant la confidentialité des données individuelles.

Outils	Algorithme	Modèles d'apprentissage au niveau des clients	les algorithme d'agrégation au niveau de serveur
IBM Federated Learning	XGBoost, Nave Bayes, Deep Reinforcement Learning, FedAvg, SPAHM, PFNM, Krum et Zeno	Réseaux neuronaux, Arbres de décision, XGBoost, Classificateurs linéaires, Naive Bayes, Apprentissage par renforcement profond	Federated Average (FedAvg), Krum, Zeno
FedML	FedAvg, FedProx, FedMA, FedNova, FedBoost, FedOpt	Federated Averaging (FedAvg), Decentralized FL, Vertical Federated Learning (VFL), Adaptive Federated Optimizer, FedNova, FedProx, FedMA	Federated Averaging (FedAvg), Federated Stochastic Gradient Descent (FedSGD), Federated Momentum, Federated Proximal, Federated Adaptive Gradient, Federated Newton, Federated QSGD, Federated Averaging with Local Adversarial Robustness
Flower	FedAvg, FedProx , QFedAvg , FedOptim	Réseaux neuronaux, Arbres de décision, Classificateurs linéaires, Naive Bayes, Apprentissage par renforcement profond	FedAvg, FedProx , QFedAvg , FedOptim
TensorFlow Federated	FedAvg, Optimisation Fédérée, k-menas, Apprentissage de Modèles de Langage Fédéré	Apprentissage fédéré pour les Données de Santé	FederatedAveraging, Federated Weighted Median, Secure Aggregation, Differential Privacy Aggregation, Custom Aggregation

TABLE 5 – Description des outils

Outils	VFL	HFL	IoT/Mobile
IBM Federated Learning	✓	✓	X
FedML	✓	✓	✓
Flower	✓	✓	✓
TensorFlow Federated	✓	✓	X

TABLE 6 – de outils Federated Learning

3 Conclusion

L'apprentissage fédéré est une approche de plus en plus utilisée dans le domaine de l'apprentissage machine, permettant de former des modèles sur des données distribuées sans avoir à les centraliser. Cependant, l'application de cette méthode à des modèles mathématiques comportant des équations et des algorithmes soulève plusieurs problématiques spécifiques.

1. Gestion des équations distribuées : Lorsque les équations d'un modèle mathématique sont distribuées sur différents appareils ou serveurs, la coordination de la résolution des équations devient complexe. Il est nécessaire de mettre en place des mécanismes de synchronisation et de communication efficaces pour assurer la convergence vers une solution globale.

2. Sécurité et confidentialité des données : Les données utilisées pour entraîner le modèle sont souvent sensibles et confidentielles. Il est crucial de garantir la sécurité et la confidentialité de ces données tout au long du processus d'apprentissage, en mettant en place des protocoles de chiffrement et d'authentification robustes.
3. Hétérogénéité des données et des appareils : Les données détenues par différentes entités peuvent être hétérogènes en termes de distribution, de format et de qualité. De plus, les appareils sur lesquels ces données résident peuvent varier en termes de puissance de calcul, de connectivité et de ressources disponibles. Gérer cette hétérogénéité de manière efficace pour garantir des performances optimales du modèle.
4. Communication et coordination : L'apprentissage fédéré nécessite une communication étroite et une coordination entre les différentes entités participantes. La mise en œuvre d'algorithmes de coordination efficaces tout en minimisant la quantité de données échangées constitue un défi complexe. De plus, la gestion des éventuelles défaillances ou des interruptions de communication nécessite une attention particulière pour assurer la robustesse du processus d'apprentissage.
5. Biais et équité : Lorsque les données sont réparties entre plusieurs entités, il est essentiel de s'assurer que le processus d'apprentissage ne renforce pas les biais existants ou n'introduise pas de nouvelles inégalités. La conception d'algorithmes d'apprentissage fédéré qui prennent en compte ces considérations est un défi important pour garantir l'équité et la représentativité des modèles résultants.

Références

[1] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*. ACM, pp. 1–8. [Online]. Available : https://dl.acm.org/doi/10.1145/3286490.3286559

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks : A comprehensive survey." [Online]. Available : http://arxiv.org/abs/1909.11875

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1273–1282, ISSN : 2640-3498. [Online]. Available : https://proceedings.mlr.press/v54/mcmahan17a.html

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning." [Online]. Available : http://arxiv.org/abs/1912.04977

[5] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," vol. 7, no. 7, pp. 5986–5994. [Online]. Available : https://ieeexplore.ieee.org/document/8917724/

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks."

[7] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization." [Online]. Available : http://arxiv.org/abs/2003.00295

[8] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, "FedSDG-FS : Efficient and secure feature selection for vertical federated learning." [Online]. Available : http://arxiv.org/abs/2302.10417

[9] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning." [Online]. Available : http://arxiv.org/abs/1905.10497

[10] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers." [Online]. Available : http://arxiv.org/abs/1912.00818

Références

[1] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*. ACM, pp. 1–8. [Online]. Available : https://dl.acm.org/doi/10.1145/3286490.3286559

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks : A comprehensive survey." [Online]. Available : http://arxiv.org/abs/1909.11875

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1273–1282, ISSN : 2640-3498. [Online]. Available : https://proceedings.mlr.press/v54/mcmahan17a.html

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning." [Online]. Available : http://arxiv.org/abs/1912.04977

[5] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," vol. 7, no. 7, pp. 5986–5994. [Online]. Available : https://ieeexplore.ieee.org/document/8917724/

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks."

[7] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization." [Online]. Available : http://arxiv.org/abs/2003.00295

[8] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, "FedSDG-FS : Efficient and secure feature selection for vertical federated learning." [Online]. Available : http://arxiv.org/abs/2302.10417

[9] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning." [Online]. Available : http://arxiv.org/abs/1905.10497

[10] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers." [Online]. Available : http://arxiv.org/abs/1912.00818