MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

MELLAH Ali Ala Eddine

THEME :

DÉTECTION DES PATHOLOGIES À BASE D'APPRENTISSAGE FÉDÉRÉ

Soutenu le :

Devant le jury composé de :

HAMAMI Dalila	Université de Mostaganem	Président
DEDDOUCHE Yamina	Université de Mostaganem	Examinateur
BENAMEUR Abdelkader	Université de Mostaganem	Encadreur

Année Universitaire 2023-2024

Remerciements

En préambule à ce travail,

Pour nous avoir permis d'être ce que nous sommes devenus aujourd'hui, pour la force et le courage qu'il nous a donné afin de terminer ce travail, nous remercions le SEIGNEUR des mondes par qui tout est possible, Allah le tout Miséricordieux.

Nous tenons à adresser notre gratitude et nos sincères remerciements à notre encadrant, **Monsieur BENAMEUR Abdelkader**. Nous le remercions pour la confiance qu'il nous a témoignée en nous proposant ce sujet, sa disponibilité, sa patience, ses conseils et directives très instructives. Nous le remercions énormément pour l'autonomie qu'il nous a laissée tout en nous aiguillant sur des prises de réflexions riches et porteuses.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner et de l'enrichir par leurs propositions.

Dédicaces

À mes chers parents pour leur amour inconditionnel,

Une pensée spéciale pour mon père dont les enseignements et les conseils continuent de me guider où que j'aille indépendamment de sa présence physique, que Dieu t'accueille dans son vaste paradis,

À ma mère chérie pour sa sagesse, sa patience et les nombreux sacrifices qu'elle a faits par amour pour moi. Je m'engage à te faire vivre la vie heureuse que tu mérites de vivre, que Dieu te garde pour nous,

À ma merveilleuse famille pour son soutien constant, notamment mes sœurs : **Fatima**, **Khadidja**, et **Samira**. Vous êtes bien plus qu'une simple famille, vous êtes mes complices, mes confidentes et mes amies pour la vie. Que notre lien spécial continue de grandir et de s'épanouir, car vous êtes les personnes les plus précieuses à mes yeux. Vous pourrez toujours compter sur moi peu importe les circonstances qui se présentent à vous,

À mes amis pour leur présence à mes côtés et dont je serai à jamais reconnaissant,

À mes enseignants pour le savoir qu'ils m'ont inculqué au cours de ces années,

Je vous remercie tous du fond du cœur.

Alaa

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes

CFL	*Centralized Federated Learning*
CH	*Configuration Hybride*
CL	*Centralized Learning*
CNN	*Convolutional Neural Network*
CP	*Configuration Physique*
CRUD	*Create, Read, Update and Read*
CSS	*Cascading Style Sheets*
CT	*Computed Tomography*
CV	*Configuration Virtuelle*
DFL	*Decentralized Federated Learning*
DL	*Deep Learning*
DP	*Differential Privacy*
DRL	*Deep Reinforcement Learning*
FedAvg	*Federated Averaging*
FL	*Federated Learning*
FLS	*Federated Learning System*
FN	*False Negative*
FP	*False Positive*
GAN	*Generative Adversial Network*
GBDT	*Generative Boosting Decision Tree*
HE	*Homomorphic Encryption*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
IA	*Intelligence Artificielle*

IRM *Imagerie par résonance magnétique*

JS *JavaScript*

ML *Machine Learning*

MLP *Multi-Layer Perceptron*

MVC *Model Vue Controller*

NN *Neural Network*

RNN *Recurrent Neural Network*

SMPC *Secure Multiparty Computation*

SPOF *Single Point Of Failure*

SVM *Support Vector Machine*

TN *True Negative*

TP *True Positive*

Introduction générale

L'intelligence artificielle (IA) est l'une des innovations les plus prometteuses de notre époque. Elle permet d'améliorer les processus dans divers secteurs, notamment celui de la santé, en offrant des solutions précises et efficaces. Depuis ses débuts, l'IA a évolué de manière spectaculaire, passant de simples algorithmes à des systèmes complexes capables de traiter des volumes massifs de données.

L'application de l'IA en médecine offre une solution pour répondre efficacement aux problèmes de difficulté variable en un temps considérablement réduit par rapport à un expert humain. Différentes techniques et algorithmes d'IA sont disponibles pour réduire l'effort des experts humains en les remplaçant dans diverses tâches. Cette approche présente un potentiel considérable pour améliorer l'efficacité des processus, la précision et la productivité des entreprises et des organisations.

Cependant, la donnée médicale, considérée comme secrète, confidentielle et privée, nécessite une protection rigoureuse. De nouvelles techniques ont été proposées pour répondre à cette préoccupation, telles que la cryptographie, le chiffrement et la compression. Parmi ces techniques, l'apprentissage fédéré se distingue en permettant la construction de modèles d'IA tout en préservant la confidentialité des données utilisées. Ce nouveau paradigme d'apprentissage repose sur le partage des poids des modèles plutôt que sur celui des données brutes.

L'évolution rapide des technologies de traitement des données transforme également le domaine de la santé. La détection précoce des maladies, notamment la pneumonie, est essentielle pour améliorer les résultats des patients. Traditionnellement, le diagnostic de la pneumonie repose sur des examens cliniques et radiologiques, nécessitant une expertise considérable. L'apprentissage profond, et en particulier les réseaux de neurones convolutifs (CNN), améliore l'automatisation et la précision des diagnostics à partir d'images médicales. Cependant, l'entraînement de ces modèles requiert de grandes quantités de données sensibles, posant des défis en matière de confidentialité.

Pour ce faire, une multitude de solutions ont été proposées pour permettre la manipulation de ce type de données en toute sécurité. Nous nous intéressons particulièrement à

l'impact de l'apprentissage fédéré, un outil permettant d'entraîner des modèles intelligents sous condition que la confidentialité des données utilisées soit garantie. L'apprentissage fédéré est un nouveau paradigme d'apprentissage profond qui a attiré la curiosité des chercheurs, qui veulent en savoir plus sur ses capacités et ses utilisations. Il s'agit d'une technique d'intelligence artificielle distribuée qui coordonne plusieurs clients avec un serveur central pour effectuer l'apprentissage sans partage de données. Par conséquent, plusieurs objets peuvent agir en tant que clients pour communiquer avec un serveur. Autrement dit, c'est le modèle qui va vers les données et non l'inverse comme cela est le cas pour les approches d'apprentissage centralisé.

Notre travail se concentre sur le développement d'un systeme générique dédiée à la détéction des pneumonia, pouvant être déployée au sein d'un hôpital. Cee systeme vise à être un outil puissant de collecte et d'aide à la décision médicale de manière efficace et sécurisée, en utilisant l'apprentissage fédéré pour assurer la confidentialité des données.

Le reste de ce rapport est structuré comme suit :

- Dans le **Chapitre I**, nous abordons dans ce chapitre l'utilisation de l'IA en médecine, ses différentes catégories et méthodologies complexes, en soulignant leur impact sur le diagnostic, les traitements personnalisés, et la gestion des soins de santé, tout en traitant les défis éthiques et pratiques.
- Dans le **Chapitre II**, nous effectuons une synthèse des travaux sur l'apprentissage fédéré. Nous présentons, d'abord, les notions de l'apprentissage automatique et l'apprentissage profond. Nous détaillons, par la suite, les différentes techniques de l'apprentissage fédéré, ses taxonomies ainsi que ses applications dans différents domaines.
- Dans le **Chapitre III**, Présente les résultats issus d'une étude expérimentale menée pour la détection de pneumonia au moyen des techniques d'apprentissage fédéré. L'objectif est de réaliser une analyse statistique de cette approche d'apprentissage.

Nous présentons, à la fin de ce rapport, la conclusion générale où nous résumons l'ensemble des points cités ainsi que d'éventuelles perspectives en vue de développer notre travail.

Chapitre I

TECHNIQUES ET APPLICATIONS MÉDICALES DE L'IA

I.1 Introduction

Dans un monde où la technologie évolue à un rythme accéléré, l'intelligence artificielle (IA) se distingue comme une révolution dans de nombreux domaines, notamment celui de la santé. Ce chapitre aborde en profondeur l'emploi de l'IA en médecine, débutant par un examen des différentes catégories d'IA—apprentissage supervisé, non supervisé et par renforcement—et s'étendant aux méthodologies complexes de l'apprentissage automatique et profond. Ces technologies révolutionnent les pratiques médicales traditionnelles, en offrant des méthodes améliorées pour le diagnostic, la personnalisation des traitements et la gestion globale des soins de santé.

Nous explorerons les diverses approches d'apprentissage en IA et comment elles peuvent être intégrées dans les systèmes de santé pour affiner les diagnostics et optimiser les traitements. En discutant des applications actuelles et potentielles de l'IA, nous illustrerons son rôle transformateur en médecine. Le chapitre traitera également des défis éthiques et pratiques associés à l'IA en santé, mettant en lumière comment des approches telles que l'apprentissage fédéré peuvent aider à surmonter ces obstacles en préservant la confidentialité des données tout en exploitant l'efficacité de l'IA collaborative.

I.2 Contexte général de l'intelligence artificielle

L'intelligence artificielle (*Intelligence Artificielle* (IA)) est un domaine de l'informatique qui vise à développer des systèmes capables de reproduire certains aspects de l'intelligence humaine, tels que le raisonnement, l'apprentissage, la planification et la résolution de problèmes complexes [2]. Le principal objectif de l'IA est de créer des machines ou des programmes informatiques intelligents qui peuvent s'adapter à différentes situations et prendre des décisions de manière autonome. Les algorithmes d'apprentissage sont apparus pour remédier à des problèmes de décision complexes et imprévisibles, les programmeurs ne peuvent pas nécessairement anticiper toutes les situations possibles ou fournir des solutions optimales à l'avance. Les algorithmes d'apprentissage permettent aux systèmes informatiques d'apprendre de manière autonome à partir de données et d'expérience, et ainsi de prendre des décisions plus précises et efficaces dans des situations variées. Ils font partie des techniques clés de l'intelligence artificielle et ont ouvert la voie à de nombreuses applications [2]. La motivation derrière le développement de l'IA est multiple. D'une part, elle permet d'automatiser des tâches complexes qui seraient fastidieuses ou impossibles à réaliser par des humains. D'autre part, les systèmes d'IA peuvent analyser et traiter de grandes quantités de données de manière plus efficace que les êtres humains. Enfin, l'IA offre la possibilité de prendre des décisions plus objectives et rationnelles, en s'appuyant

sur des algorithmes et des règles prédéfinies [2].

I.2.1 Les catégories de l'intelligence artificielle

On distingue généralement trois grandes catégories d'IA [2] :

— **L'apprentissage supervisé**,C'est la forme la plus courante, où un modèle apprend à partir d'exemples annotés pour prédire des résultats pour de nouvelles données. Par exemple, en médecine, l'apprentissage supervisé permet de prédire si une image médicale montre des signes de maladie.

— **L'apprentissage non supervisé**, Ici, le modèle cherche des motifs ou des structures dans des données non étiquetées. Les applications incluent la segmentation de marché en marketing et l'identification de groupes de gènes en génétique.

— **L'apprentissage par renforcement**, où le système apprend par essais et erreurs en interagissant avec un environnement pour maximiser une récompense à long terme.

I.3 Apprentissage automatique

L'apprentissage automatique (*Machine Learning* (ML)) est une application de l'intelligence artificielle qui permet à l'agent intelligent d'inférer des résultats plus précis sans avoir à les programmer explicitement. Il s'appuie sur la création des algorithmes capables de recevoir des données d'entrée et d'utiliser une analyse statistique pour prédire une sortie.

L'apprentissage automatique est subdivisé en trois (03) catégories selon la description et l'annotation des données d'apprentissage [3]. Nous définissons dans ce qui suit chacune de ces catégories.

I.3.1 Apprentissage supervisé

Quand les données d'apprentissage sont étiquetées et l'espace d'apprentissage est fermé, nous disons que l'apprentissage est supervisé [4]. En d'autres termes, les experts du domaine d'application ont annoté les données pour qu'elles soient exploitables. Le but est de pouvoir affecter une nouvelle observation à une classe déterminée. L'algorithme permet donc d'établir les frontières entre les différentes classes.

Les algorithmes d'apprentissage supervisé sont applicables dans les problèmes de classification (Plus proches voisins, Machines à vecteurs de support(*Support Vector Machine* (SVM)), Réseaux de neurones convolutifs, etc.) et de régression (Régression linéaire, Arbres de décision, etc.).

I.3.2 Apprentissage non supervisé

Dans les situations auxquelles l'agent intelligent n'a aucune connaissance sur la nature d'une instance ou son appartenance à une classe spécifique, l'apprentissage est dit non supervisé. La tâche principale de l'algorithme est d'établir d'abord une topologie qui guide les observations pour permettre de classer les objets et du coup regrouper les instances proches en groupes appelés clusters [4].

Le partitionnement de données (*clustering*) est la méthode la plus utilisée dans ce type de problèmes. L'idée est d'organiser les données brutes (non annotées) en partitions. Les données de chaque sous-ensemble partagent des caractéristiques communes relatives dans la majorité des cas à des mesures de proximité. Il existe de multiples méthodes de partitionnement de données, parmi lesquelles nous citons l'algorithme des K-moyennes et le partitionnement hiérarchique.

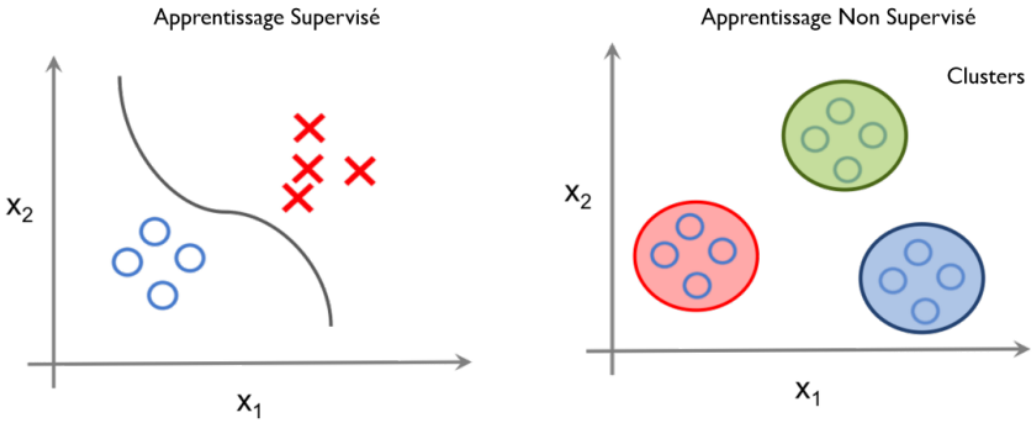


Figure I.1 — Différence entre apprentissage supervisé et apprentissage non supervisé. [4].

I.3.3 Apprentissage par renforcement

L'apprentissage par renforcement est le problème auquel est confronté un agent qui doit apprendre son comportement en interagissant avec son environnement dynamique. Il se base sur les interactions essais-erreurs [3]. C'est-à-dire, l'agent n'a au préalable aucune information sur l'environnement ni sur ce qu'il peut y être. Il commence ainsi à explorer son entourage pour pouvoir extraire des connaissances qui l'aident à établir le modèle de calcul.

Bien que les méthodes d'apprentissage machine traditionnelles aient connu un grand succès, elles présentent certaines limitations, notamment en matière de traitement de données complexes et non structurées comme les images, les signaux ou les textes. De plus, elles nécessitent souvent une étape d'extraction manuelle des caractéristiques pertinentes,

ce qui peut être un processus chronophage et dépendant du domaine [5]. C'est dans ce contexte que l'apprentissage profond (deep learning) est apparu comme une approche révolutionnaire pour surmonter ces défis.

Technique	**Avantages**	**Inconvénients**
Apprentissage supervisé	Bon pour les tâches avec des données étiquetées abondantes. Modèles prédictifs précis. Largement utilisé et bien compris.	Nécessite des données étiquetées, qui peuvent être coûteuses et difficiles à obtenir. Moins efficace pour les problèmes complexes ou non linéaires. Peut conduire à des modèles biaisés si les données d'apprentissage ne sont pas représentatives.
Apprentissage non supervisé	Ne nécessite pas de données étiquetées. Peut identifier des structures cachées dans les données. Utile pour l'exploration de données et la détection d'anomalies.	Peut être difficile à interpréter les résultats. Les performances peuvent varier en fonction de la complexité des données. Nécessite plus de données que l'apprentissage supervisé.
Apprentissage par renforcement	Peut apprendre à partir d'interactions avec l'environnement. Adaptable à des problèmes complexes et dynamiques. Peut apprendre des stratégies optimales sans programmation explicite.	Peut nécessiter beaucoup d'essais et d'erreurs pour apprendre. Peut être difficile à optimiser l'algorithme d'apprentissage. Les actions de l'agent peuvent avoir des conséquences imprévues dans l'environnement réel.

Table I.1 — Comparaison des techniques d'apprentissage automatique

I.4 Apprentissage profond

L'apprentissage profond (*Deep Learning* (DL)) est une partie de l'apprentissage automatique. Il repose sur la notion de réseaux de neurones artificiels qui représentent un outil pour faire de l'apprentissage supervisé. Ces réseaux de neurones s'appuient sur un modèle mathématique inspiré de l'architecture biologique du cerveau humain. Ils se constituent d'une collection de nœuds connectés entre eux, appelés neurones [6].

Il existe plusieurs types de réseaux de neurones artificiels, parmi lesquels nous citons le perceptron multicouche et les réseaux de neurones convolutifs.

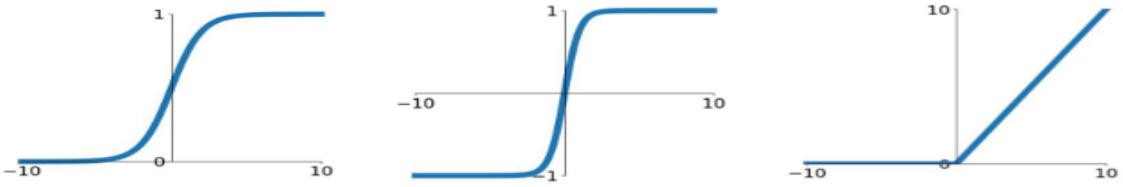
I.4.1 Perceptron multicouche

Le perceptron multicouche (*Multi-Layer Perceptron* (MLP)) a été inventé en 1957 par Frank Rosenblatt [6]. Il est organisé en plusieurs couches au sein desquelles un signal ou une information circule de la couche d'entrée vers la couche de sortie uniquement. Il s'agit donc d'un réseau à propagation directe. Chaque couche du réseau peut contenir un ensemble variable de neurones.

I.4.2 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (*Convolutional Neural Network* (CNN)) consistent en un empilage de couches Perceptron pour le pré-traitement de petites quantités d'informations [7]. Ils se composent d'un ensemble de couches qui sont [8] :

- **Couche de convolution** : cette couche représente le bloc de base pour la construction des CNN. Elle est dimensionnée par trois hyper-paramètres (la profondeur, le pas et la marge).
- **Couche de pooling** : elle permet la compression des données (généralement images) en réduisant leurs tailles. Il en existe plusieurs types à savoir *Max Pooling* et *Average Pooling*.
- **Couche de correction** : cette couche est utilisée généralement pour rendre le traitement plus efficace, et ce, en intercalant entre les couches de traitement une couche qui va exercer une fonction d'activation sur la sortie. La Fig. I.2 montre quelques fonctions utilisées pour la correction.



(a) Fonction Sigmoïde (b) Tangente hyperbolique (c) Unité linéaire rectifiée

Figure I.2 — Exemples de fonctions d'activation utilisées dans la couche de correction [9].

- **Couche entièrement connectée** : dans cette couche illustrée dans la Fig. I.3, tous les nœuds contenus se connectent à tous les nœuds de la couche suivante. Elle se met généralement vers la fin des réseaux de neurones pour classer les données en différentes classes.

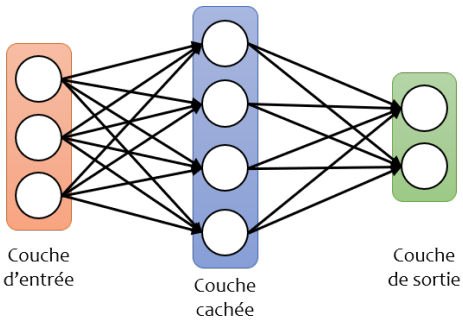


Figure I.3 — Couche entièrement connectée [10].

I.5 Application de l'IA dans le domaine médical

L'intégration de l'intelligence artificielle en médecine transforme le secteur de la santé grâce à sa capacité à améliorer les diagnostics, les plans de traitement et les résultats pour les patients. Le rôle de l'IA s'étend des processus analytiques aux applications pratiques en milieu clinique, ayant un impact significatif sur les pratiques médicales et les soins aux patients.

I.5.1 Applications actuelles de l'IA dans les diagnostics et traitements médicaux

Les technologies IA ont prouvé leur importance dans le diagnostic de maladies avec une précision supérieure à celle des méthodes traditionnelles. Par exemple, les algorithmes IA peuvent analyser des données médicales complexes et reconnaître des motifs imperceptibles à l'œil humain. Cette capacité est particulièrement précieuse dans des domaines comme la radiologie, la pathologie et la dermatologie où l'imagerie précise est cruciale. [11]

I.5.1.1 Études de cas et intégration technologique

— **Diagnostic clinique :** Les systèmes IA sont utilisés pour détecter des conditions telles que le cancer, les troubles neurologiques et les maladies cardiovasculaires en évaluant les images médicales, les informations génétiques et les données des patients de manière plus rapide et précise. [12]

— **Optimisation du traitement :** L'IA est utilisée pour prédire les réponses des patients à différents traitements, permettant ainsi de personnaliser les thérapies selon les besoins individuels, ce qui améliore l'efficacité des interventions médicales.

I.5.2 Croissance et potentiel de l'IA dans le secteur de la santé

Le potentiel de l'IA dans le secteur de la santé est vaste, avec des avancées continues qui ouvrent de nouvelles voies pour la recherche médicale et l'application. L'IA est

prête à révolutionner la prestation des soins de santé en permettant une médecine plus personnalisée, en réduisant les coûts des soins de santé et en améliorant la qualité des soins [13].

I.5.2.1 Amélioration des systèmes de santé

L'IA aide à rationaliser les processus administratifs dans les établissements de santé, de la prise de rendez-vous à la gestion des dossiers des patients, augmentant ainsi l'efficience opérationnelle et permettant au personnel médical de se concentrer davantage sur les soins aux patients [14].

I.5.3 Défis et considérations éthiques

Malgré ses avantages, le déploiement de l'IA en médecine est accompagné de défis, y compris des préoccupations éthiques autour de la vie privée, du consentement et de la fiabilité des systèmes IA. Assurer que ces technologies sont utilisées de manière responsable et transparente est crucial pour leur succès et leur acceptation dans le domaine de la santé.

L'intelligence artificielle en médecine offre des améliorations prometteuses aux services de santé, en améliorant les diagnostics, la gestion des patients et les protocoles de traitement. Au fur et à mesure que l'IA continue d'évoluer, elle devrait devenir une partie intégrante de la santé, remodelant le paysage médical pour le rendre plus efficace et centré sur le patient [15].

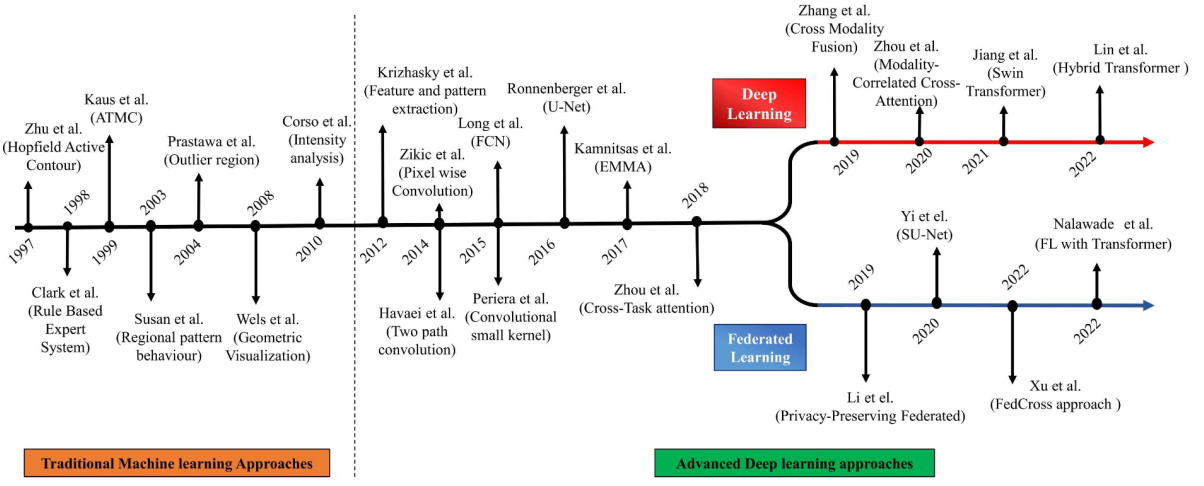


Figure I.4 — Évolution des méthodes en intelligence artificielle dans le domaine de la santé [1].

I.6 Topologies d'apprentissage dans l'IA

Le domaine de l'intelligence artificielle (IA) a connu un essor considérable ces dernières années, en grande partie grâce aux avancées dans les techniques d'apprentissage automatique. Cependant, la manière dont les données sont collectées, stockées et utilisées pour entraîner les modèles d'IA a un impact significatif sur les performances, la sécurité et la confidentialité des systèmes résultants. Différentes topologies d'apprentissage ont émergé pour répondre à ces préoccupations, chacune présentant ses propres avantages et défis.

I.6.1 Apprentissage centralisé

L'apprentissage centralisé est la topologie la plus traditionnelle, dans laquelle toutes les données sont rassemblées et stockées dans un endroit central, généralement un serveur ou un centre de données [16]. Un modèle unique est alors entraîné sur cet ensemble de données complet. . Cette approche centralisée offre plusieurs avantages, notamment :

— L'accès à un volume important de données, ce qui peut améliorer les performances du modèle.

— La simplicité de mise en œuvre et de maintenance, avec un point de contrôle unique.

— La possibilité d'appliquer des techniques d'apprentissage avancées nécessitant des ressources de calcul importantes.

Cependant, l'apprentissage centralisé soulève des préoccupations majeures en termes de confidentialité et de sécurité des données, en particulier dans des domaines sensibles comme la santé. Le transfert et le stockage de données personnelles dans un endroit centralisé augmentent les risques de violations de données et de cyberattaques. De plus, cette approche peut être coûteuse en termes de bande passante et de stockage, en particulier lorsque les données sont volumineuses ou distribuées géographiquement.

I.6.2 Apprentissage distribué

L'apprentissage distribué est une topologie dans laquelle les données et les ressources de calcul sont réparties sur plusieurs nœuds ou sites distants. Chaque nœud entraîne un modèle local sur ses propres données, puis ces modèles locaux sont agrégés pour former un modèle global. Cette approche offre plusieurs avantages [17] :

— Une meilleure évolutivité, car le calcul est parallélisé sur plusieurs nœuds.

— Une certaine préservation de la confidentialité, car les données ne sont pas transférées de manière centralisée.

— Une réduction des coûts de stockage et de transfert de données.

Cependant, l'apprentissage distribué présente également des défis importants, notamment :

— La nécessité d'une coordination complexe entre les nœuds pour synchroniser l'entraînement et l'agrégation des modèles.

— Des problèmes potentiels liés à la non-indépendance et à la non-identité des distributions de données (Non-IID) entre les nœuds.

— Des besoins en mécanismes de sécurité robustes pour protéger les communications entre les nœuds.

I.6.3 Apprentissage fédéré

L'apprentissage fédéré est une approche décentralisée proposée récemment pour relever les défis de confidentialité et de sécurité des données. Dans cette topologie, les données restent locales sur chaque nœud ou dispositif et ne sont jamais échangées ni centralisées. Les modèles sont entraînés localement sur chaque nœud, puis les mises à jour des modèles sont agrégées de manière sécurisée pour former un modèle global partagé [18].

L'apprentissage fédéré présente plusieurs avantages clés :

— La préservation de la confidentialité des données, car celles-ci ne quittent jamais leur emplacement d'origine.

— Une meilleure sécurité, car seules les mises à jour des modèles sont échangées, réduisant les risques de violations de données.

— La possibilité de tirer parti de la puissance de l'apprentissage collaboratif tout en respectant les réglementations sur la protection des données.

I.7 Conclusion

Ce chapitre a exploré les multiples dimensions de l'intelligence artificielle et son intégration progressive mais déterminante dans le secteur médical. À travers l'examen des différentes formes d'apprentissage automatique, nous avons mis en lumière comment chaque méthode contribue à façonner les outils de diagnostic et de traitement modernes. Les applications de l'IA dans la médecine ne se limitent pas seulement à améliorer les capacités diagnostiques et thérapeutiques, mais s'étendent également à la personnalisation des soins, offrant ainsi des traitements mieux adaptés aux besoins individuels des patients.

Nous avons également abordé les défis inhérents à l'adoption de l'IA dans un domaine aussi sensible que la médecine, notamment les considérations éthiques et les questions de confidentialité des données. Ces obstacles nécessitent des solutions innovantes pour garantir que le déploiement de l'IA soit bénéfique et équitable pour tous les acteurs concernés.

En conclusion, l'intelligence artificielle représente un potentiel transformateur pour le domaine médical. En tirant parti de l'apprentissage fédéré et d'autres avancées en IA, le chapitre suivant approfondira notre compréhension de l'apprentissage fédéré, en explorant en détail ses principes, ses applications pratiques et son potentiel pour révolutionner davantage le domaine médical.

Chapitre II

ÉTAT DE L'ART SUR L'APPRENTISSAGE FÉDÉRÉ

II.1 Introduction

L'application de l'intelligence artificielle (IA) offre une solution pour répondre efficacement aux problèmes de difficulté variable en un temps considérablement réduit par rapport à un expert humain. Différentes techniques et algorithmes d'IA sont disponibles pour réduire l'effort de l'expert humain en le remplaçant dans diverses tâches. Cette approche présente un potentiel considérable pour améliorer l'efficacité des processus, la précision et la productivité des entreprises et des organisations.

De plus, la donnée nécessite d'être confidentielle. C'est pour cela que sa protection est une préoccupation majeure dans de nombreux domaines. Pour y répondre, de nouvelles techniques ont été proposées, telles que la cryptographie, le chiffrement, la compression, entre autres. Parmi ces techniques, l'apprentissage fédéré se distingue en permettant la construction de modèles d'intelligence artificielle tout en préservant la confidentialité des données utilisées. Ce nouveau paradigme d'apprentissage se base sur le partage des poids des modèles plutôt que sur celui des données brutes. Il s'agit d'un paradigme fondamental sur lequel se base notre travail, raison pour laquelle nous lui consacrons ce deuxième chapitre.

De ce fait, nous présentons dans ce chapitre divers concepts de base qui permettent d'appréhender ce nouveau paradigme. Nous abordons le concept d'apprentissage fédéré en soulignant son importance, ainsi que son application dans divers domaines, les défis qu'il permet de relever et leurs solutions. Nous présentons, par ailleurs, une taxonomie pour distinguer les différents types de systèmes d'apprentissage fédéré. Enfin, nous donnons un aperçu sur les travaux d'application de l'apprentissage fédéré dans le domaine médical.

II.2 Apprentissage fédéré

L'apprentissage fédéré (*Federated Learning* (FL)) est une technique d'apprentissage automatique qui entraîne un algorithme sur plusieurs clients décentralisés. Chacun de ces clients détient un ensemble local de données sur lequel il entraîne le modèle. En ce sens, ces ensembles de données sont privés et ne sont pas échangés entre les clients, seuls les poids issus du processus d'apprentissage le sont [19]. Dans cette section, nous présentons les motivations qui ont conduit à introduire l'apprentissage fédéré. Par la suite, nous décrivons son processus ainsi que quelques domaines de son application.

II.2.1 Historique et motivations de l'apprentissage fédéré

L'apprentissage fédéré a été introduit par la compagnie Google en 2016 pour la première fois pour améliorer son propre modèle de saisie sur les téléphones Android, appelé *Gboard*,

sans avoir à télécharger les données sensibles du clavier des utilisateurs [19]. L'idée était de permettre aux utilisateurs d'entraîner chacun son modèle de saisie et puis d'envoyer les paramètres des modèles vers le *cloud* pour les fusionner.

Cette approche s'oppose aux techniques d'apprentissage centralisé où l'entraînement se fait au niveau d'un serveur central où tous les clients lui envoient leurs données pour qu'il entraîne le modèle intelligent. L'apprentissage fédéré diffère ainsi des techniques d'apprentissage distribué qui permettent au client détenant un ensemble de données centralisé de distribuer l'entraînement du modèle sur plusieurs clients. Autrement dit, il a une vision globale sur toutes les données [20].

À la différence à ces techniques, l'apprentissage fédéré fait intervenir plusieurs clients connectés à un serveur. Chacun d'eux dispose de son ensemble de données local. Aucun client ou serveur n'a de connaissances sur ce que les autres clients possèdent. Le serveur est seulement responsable de stocker la base de connaissances ou bien le modèle intelligent partagé entre les différents clients. L'entraînement se fait au niveau des clients et le serveur joue dans ce cas le rôle d'un coordinateur. En d'autres termes, l'apprentissage fédéré apporte des modèles intelligents à la source de données, plutôt que d'apporter les données au modèle [21]. Il permet à de multiples acteurs de construire un modèle d'apprentissage commun et robuste sans partage de données. Ceci permet de résoudre les problèmes critiques liés à la confidentialité, la sécurité et les droits d'accès aux données [22].

II.2.2 Processus de l'apprentissage fédéré

Le processus d'un apprentissage fédéré se décompose en trois (03) principales étapes dont la préparation des paramètres d'apprentissage, le calcul des modèles locaux et l'agrégation et la mise à jour du modèle global [20]. Ce processus est illustré dans la Fig. II.1.

1. **Préparation des paramètres d'apprentissage** : dans un premier temps, le serveur sélectionne les clients qui participent à l'entraînement. Il définit en outre les hyperparamètres liés au processus d'apprentissage (nombre d'epochs, taux d'entraînement, etc.). Enfin, il diffuse ces informations ainsi que le modèle global aux clients choisis.
2. **Calcul des modèles locaux** : dans cette phase, chaque client i reçoit le modèle global et lance l'apprentissage pour avoir son modèle local dénoté ω_i^t en se basant sur son jeu de données local. L'objectif de tout client est de minimiser sa fonction de perte. Une fois le modèle local calculé, il est envoyé vers le serveur.
3. **Agrégation et mise à jour du modèle global** : le serveur, après réception de tous les modèles locaux des différents clients, procède à une opération d'agrégation pour

calculer le modèle global. Il applique un algorithme de moyennage sur l'ensemble des modèles locaux. À la fin, le serveur partage le modèle résultant avec tous les clients qui lui sont connectés.

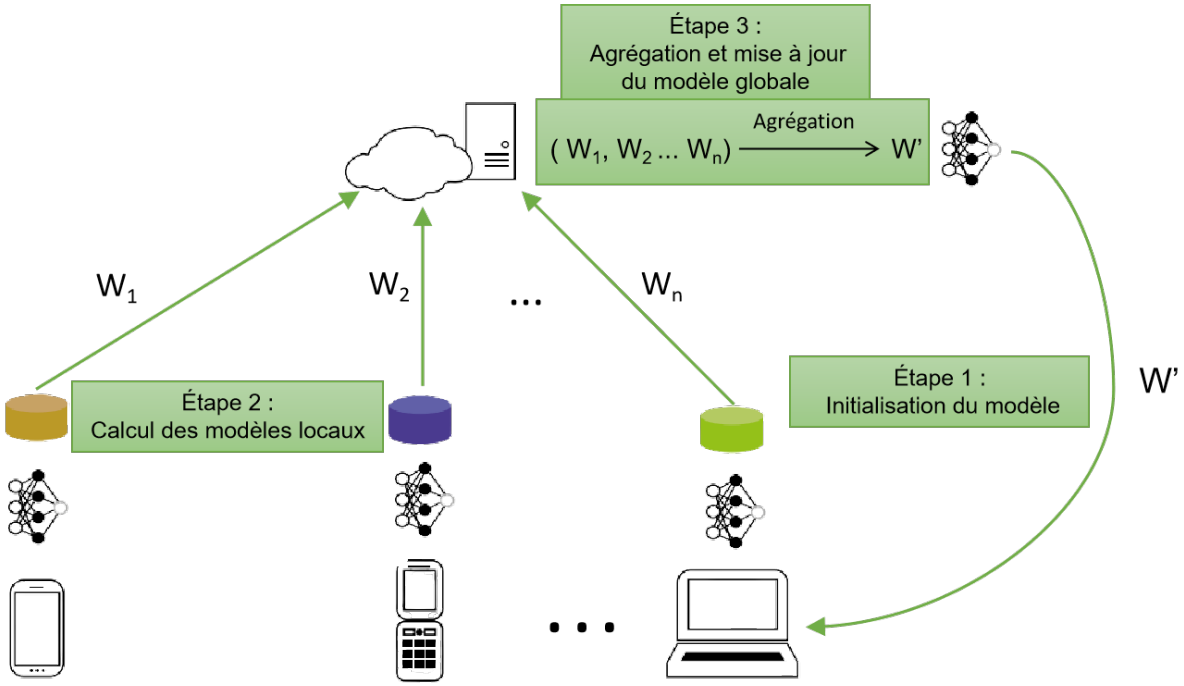


Figure II.1 — Processus d'apprentissage fédéré [23].

Le processus d'apprentissage s'arrête dans les situations suivantes :

— L'arrêt d'apprentissage est contrôlé, à la base, par le nombre de tours de communication défini dès le départ par le serveur dans la première phase.

— Le serveur déclare la terminaison du processus d'apprentissage, et ce, en envoyant un message d'arrêt aux clients participants. Cet arrêt peut être intentionnel ou involontaire (par exemple, une panne au niveau du serveur).

— Les clients participants décident de mettre fin au processus d'entraînement des modèles. La décision doit être faite par la majorité des clients pour procéder à un arrêt du processus.

— Le modèle global converge rapidement et la fonction perte ne change pas.

II.2.3 Domaines d'applications de l'apprentissage fédéré

Les applications de l'apprentissage fédéré s'étendent sur un certain nombre d'industries, notamment les télécommunications, la cybersécurité, le traitement du langage naturel, l'analyse financière et la médecine. Parmi ces travaux, nous citons :

— **Traitement automatique du langage naturel** : Hard et al. [24] appliquent l'apprentissage fédéré à la prédiction du mot suivant sur clavier mobile. Ils adoptent la

méthode de moyenne fédérée pour améliorer une variante de réseau de neurone appelée (*Coupled Input and Forget Gate*). La méthode d'apprentissage fédéré peut obtenir une meilleure précision de rappel que l'entraînement basé sur un serveur avec des données se trouvant dans un fichier *log*.

— **Détection de fraude** : Zheng et al. [25] introduisent l'apprentissage fédéré dans le domaine de la détection de fraude sur les transactions de cartes de crédit. Ils conçoivent un nouveau *framework* d'apprentissage fédéré basé sur un méta-apprentissage, appelé réseau K-tuplet profond, qui garantit non seulement la confidentialité des données, mais qui obtient également des performances significativement supérieures par rapport aux approches existantes.

— **Systèmes de recommandations** : Ammad-ud din et al. [26] formulent la première méthode de filtrage collaboratif fédéré. Sur la base d'une approche gradient stochastique, la matrice facteur d'éléments est formée sur un serveur global en agrégeant les mises à jour locales. Ils montrent empiriquement que la méthode fédérée n'a presque aucune perte de précision par rapport à la méthode centralisée.

Chai et al. [27] conçoivent un *framework* de factorisation de matrices fédéré. Ils utilisent l'algorithme du gradient stochastique fédéré pour entraîner les matrices. De plus, ils adoptent le chiffrement homomorphe pour protéger les gradients communiqués.

— **Domaine médical** : Pour la détection du cancer du sein, Jimenez et al. [28] ont développé un système collaboratif de diagnostic assisté par ordinateur. Ce système qui se base sur l'apprentissage fédéré, ne repose pas sur des données de la même modalité, mais qui fait intervenir des clients avec des données hétérogènes et qui ne sont pas identiquement distribuées.

II.3 Défis et solutions de l'apprentissage fédéré

Il existe plusieurs problèmes évoqués dans les études sur l'apprentissage fédéré. Ces problèmes sont associés à divers aspects tels que le partage de modèles entre le serveur et les clients, l'hétérogénéité des ressources des clients, la sécurité des modèles et des données. Cette section expose les défis ainsi que les approches suggérées pour les surmonter.

II.3.1 Sécurité des données et des modèles

Malgré le fait que les données locales ne soient pas exposées lors de l'utilisation de l'apprentissage fédéré, les modèles peuvent être interceptés ou perturbés pendant la communication, ce qui peut entraîner des risques pour la confidentialité et l'intégrité des

données et des modèles échangés [29].

II.3.1.1 Types d'attaques

Dans la littérature, plusieurs types d'attaques ont été identifiés.

— **Attaques de perturbation** : dans ce type d'attaque, l'attaquant peut perturber les données envoyées par les appareils clients pour introduire du bruit dans le modèle global. Cela peut être fait en modifiant ou en ajoutant du bruit aux modèles envoyées par un client pour corrompre le modèle global. Cela peut affecter la précision et la fiabilité du modèle global [29].

— **Attaques d'inférence** : dans ce type d'attaque, les attaquants peuvent essayer d'inverser le modèle global en l'analysant. Ceci permet d'obtenir des informations sur les données d'entraînement utilisées pour le former ou pour identifier les appareils clients qui ont contribué à la formation du modèle. Cela peut compromettre la confidentialité des données d'entraînement et des appareils clients [30].

— **Attaques par empoisonnement de données** : dans ce type d'attaque, les attaquants peuvent injecter des données malveillantes dans le modèle global pour le faire dérailler ou pour orienter ses résultats. Cela peut être fait en injectant des données qui ressemblent à des données normales, mais qui ont des étiquettes erronées ou des caractéristiques qui ne sont pas représentatives de la population cible. Cela peut affecter la précision et la fiabilité du modèle global [29].

— **Attaques de substitution de modèle** : dans ce type d'attaque, les attaquants peuvent remplacer le modèle global par un modèle malveillant ou biaisé pour tromper les utilisateurs finaux ou les serveurs. Cela peut être fait en remplaçant le modèle global par un modèle qui a été entraîné avec des données biaisées pour favoriser certains utilisateurs ou groupes d'utilisateurs. Cela peut affecter la précision et la fiabilité du modèle global et peut également avoir des conséquences éthiques [31].

II.3.1.2 Solutions de confidentialité

Il existe une variété de mécanismes de confidentialité développés pour faire face aux différents types d'attaque (Section II.3.1.1), tels que les algorithmes cryptographiques, qui peuvent aider à protéger la confidentialité des données et des modèles pendant la communication. Cependant, des ressources de calcul supplémentaires sont nécessaires pour le cryptage, ce qui compromettra l'efficacité de l'entraînement du modèle [32]. Nous présentons les principales approches utilisées dans les systèmes d'apprentissage fédéré actuels pour protéger les données.

— **Confidentialité différentielle** : c'est une technique cryptographique consistant à ajouter un bruit aléatoire contrôlé à un ensemble de données sous prétexte que l'ensemble résultant soit suffisamment précis [33]. Le bruit ajouté peut suivre n'importe quelle forme de distribution probabiliste (laplacienne, gaussienne ou exponentielle).

— **Techniques cryptographiques** : les techniques cryptographiques sont utilisées pour protéger la confidentialité et l'intégrité des données pendant le processus d'entraînement du modèle, contre les attaques de vol des informations privées et de l'interception des poids, des gradients et du modèle global.

- **Chiffrement homomorphe** : il permet de réaliser des calculs directement sur des textes chiffrés, ce qui garantit que le résultat déchiffré est identique au résultat obtenu sur le texte clair correspondant. Cette méthode est largement utilisée [34, 35].
- **Calcul multipartite sécurisé** : les méthodes basées sur le calcul multipartite sécurisé permettent aux participants distribués de calculer conjointement une fonction objective sans révéler leurs propres données [36, 37].
- **Partage de secret** : il permet de diviser un secret en plusieurs parts distribuées à différents acteurs du système. Le partage de secret est utilisé dans de nombreux frameworks d'apprentissage fédéré [38, 39] pour préserver la vie privée des données des participants.

— **Techniques d'anonymisation** : les méthodes basées sur l'anonymisation sont utilisées pour protéger les données privées en les modifiant de manière à ce qu'elles ne puissent pas être identifiées comme appartenant à une personne spécifique. Cette anonymisation est effectuée avant de les utiliser pour entraîner un modèle global, ce qui protège la confidentialité des données contre les attaques de personnes malveillantes. Le schéma de k anonymisation est une technique qui permet de préserver la confidentialité des données tout en conservant leur utilité et en améliorant les performances du modèle [40].

— **Techniques hybrides** : ces techniques combinent différentes méthodes pour équilibrer le compromis entre la confidentialité des données et leur utilité. Certaines méthodes hybrides sont basées sur le chiffrement homomorphe et le partage de secret pour préserver la confidentialité des données [41], et d'autres méthodes utilisent une combinaison de calculs sécurisés multipartites et de protection de la vie privée différentielle [42] pour garantir que les informations privées ne soient pas divulguées pendant le processus de d'entraînement.

II.3.2 Communication des modèles

Dans l'apprentissage fédéré, les modèles sont entraînés sur des données décentralisées, stockées chez les clients. Les modèles sont ensuite agrégés sur un serveur central pour former un modèle global. Nous citons les différents problèmes liés à la communication :

II.3.2.1 Problèmes liés à la communication des modèles

La communication entre les clients et le serveur est un défi majeur dans l'apprentissage fédéré, car elle peut être coûteuse en termes de latence, de bande passante et de sécurité.

— **Latence** : l'un des principaux défis de l'apprentissage fédéré est la synchronisation des modèles entre les serveurs et les clients. En effet, lors de chaque itération de l'apprentissage, les modèles doivent être envoyés des clients au serveur, et inversement, ce qui peut entraîner des temps de latence importants. Cette latence peut être due à plusieurs facteurs, tels que la bande passante limitée des clients ou la complexité du modèle. La gestion efficace de ce problème de synchronisation est donc cruciale pour assurer l'efficacité de l'apprentissage fédéré [21].

— **Coût de communication** : le transfert des modèles entre les clients et le serveur peut être coûteux, en particulier lorsque le modèle est volumineux ou quand la communication doit être effectuée fréquemment [43].

— **Compromis entre l'efficacité et la confidentialité** : pour assurer la confidentialité des modèles échangés, les clients doivent crypter les paramètres des modèles avant de les renvoyer au serveur central. Par conséquent, des ressources de calcul supplémentaires sont nécessaires pour le cryptage, ce qui compromettra l'efficacité de l'entraînement du modèle [32].

II.3.2.2 Solutions développées

Afin de pallier aux problèmes de communication rencontrés dans le contexte de l'apprentissage fédéré, plusieurs méthodes ont été développées. Ces méthodes sont les suivantes :

— **Masque aléatoire** : cette technique consiste à limiter la mise à jour du modèle en ne transmettant que les gradients importants au serveur. Pour ce faire, un masque aléatoire est appliqué sur le modèle avant l'envoi des gradients. Ce masque permet de sélectionner les gradients non nuls et de les envoyer au serveur pour mettre à jour le modèle global, ce qui permet de réduire la taille de données transmises et donc la charge de communication [44].

— **Quantification des modèles** : elle consiste à compresser le modèle en réduisant la précision des nombres représentant les poids du modèle. Cette compression permet de réduire la taille du modèle avant de l'envoyer au serveur et par la même occasion le coût de communication [45].

— **Élagage de poids** : il est utilisé pour réduire la quantité de données qui doivent être transférées entre les clients et le serveur lors des étapes de communication. En supprimant les poids de faible importance, on peut réduire la taille des modèles et ainsi diminuer le coût de communication [46].

II.3.3 Hétérogénéité de clients

Dans l'apprentissage fédéré, les clients sont les possesseurs de données et jouent un rôle crucial dans l'entraînement du modèle global. Cependant, ces clients peuvent présenter des différences significatives en termes de caractéristiques de données telles que la taille des données, la qualité des données et les distributions de données [47].

II.3.3.1 Problèmes causés par l'hétérogénéité de clients

L'hétérogénéité des clients peut avoir un impact négatif sur la qualité du modèle global. De ce fait, voici quelques problèmes causés par cette hétérogénéité :

— **Déséquilibre de la distribution des données** : lors de l'utilisation de l'apprentissage fédéré, les modèles sont entraînés sur les données des clients, qui peuvent être inégalement réparties en termes de distribution et de quantité. Par exemple, un client peut avoir un ensemble de données plus important ou plus diversifié que les autres clients, ce qui peut créer un déséquilibre entre les clients [48].

  Ce déséquilibre peut entraîner des biais dans le modèle global, car le modèle sera plus influencé par les clients qui détiennent des données ayant une plus grande présence dans l'ensemble de données global. Le manque de représentativité des données peut amener le modèle à les ignorer où les mal comprendre. En revanche, si les données sont surreprésentées dans l'ensemble de données global, le modèle peut les surpondérer et ne pas généraliser correctement aux autres données, ce qui peut entraîner de mauvaises prédictions pour de nouvelles données [49].

— **Variation de la qualité des données** : certains clients peuvent avoir des données manquantes, ce qui signifie que certaines informations importantes ne figurent pas dans leur ensemble de données. D'autres clients peuvent avoir des erreurs dans leurs données, c'est-à-dire, certaines informations sont incorrectes ou mal étiquetées. En outre, il peut y avoir des problèmes de bruit dans les données, où certaines infor-

mations sont corrompues ou incomplètes. Si ces problèmes ne sont pas gérés correctement, cela peut entraîner une dégradation de la qualité du modèle global. Les données manquantes peuvent empêcher le modèle de généraliser correctement, les données incorrectes peuvent entraîner des prédictions erronées, alors que le bruit apporte un impact négatif sur la précision du modèle.

— **Variation des types de données** : les clients peuvent posséder différents types de données, par exemple, des images, des textes, des signaux audio, etc. Cela peut rendre difficile l'agrégation de ces données pour former un modèle global cohérent, car différents types de données nécessitent des techniques d'apprentissage différentes.

— **Capacités de calculs des clients** : les clients peuvent avoir des capacités de calcul différentes, ce qui peut engendrer un impact significatif sur le temps requis pour l'apprentissage du modèle global. Par exemple, un client disposant d'une grande puissance de calcul peut traiter les données plus rapidement qu'un client disposant d'une faible puissance de calcul. Si un grand nombre de clients ont des capacités de calcul limitées, cela peut considérablement ralentir le temps d'apprentissage du modèle global.

II.3.3.2 Solutions aux problèmes d'hétérogénéité des clients

Afin de résoudre les problèmes d'hétérogénéité des clients dans l'apprentissage fédéré, plusieurs méthodes ont été proposées dans la littérature. Parmi ces méthodes, certaines sont plus couramment utilisées que d'autres. On retrouve dans la littérature plusieurs solutions pour résoudre les défis de l'apprentissage fédéré. Fang et al. [50] ont récemment recensé plusieurs méthodes, parmi lesquelles nous citons les suivantes :

— **Agrégation pondérée** : cette méthode consiste à attribuer un poids à chaque client en fonction de la quantité et de la qualité de ses données. Les clients ayant une quantité ou une qualité de données plus importante auront ainsi un poids plus élevé, ce qui signifie que leurs données auront une plus grande influence sur le modèle global [51]. Elle permet de réduire le biais causé par le déséquilibre de la distribution des données et d'obtenir un modèle global plus robuste et plus représentatif de l'ensemble des données disponibles.

— **Correction de pertes** : cette technique implique le calcul de la combinaison pondérée optimale des clients participants, via une approche appelée CCR. Cette méthode s'adapte de manière dynamique en ajustant la contribution de chaque client pendant la mise à jour des pertes, en réduisant la contribution des clients bruyants et en augmentant celle des clients fiables [50].

— **Fédération horizontale** : les clients ayant des types de données similaires peuvent être regroupés pour former des sous-groupes et entraîner des modèles locaux cohérents avant l'agrégation. Cette approche est appelée fédération horizontale et peut aider à gérer la variation des types de données [52].

— **Communication comprimée** : la communication entre les clients et le serveur peut être réduite en utilisant des techniques de compression des données, telles que la compression des modèles et la quantification des mises à jour de modèles [45]. Cela peut aider à réduire l'impact des capacités de calcul différentes des clients sur le temps global d'apprentissage.

II.3.4 Sélection des clients

L'architecture d'un système fédéré comporte un serveur et plusieurs clients qui font l'échange de modèles pour une tâche de généralisation du modèle global. La première phase consiste à préparer les différents paramètres ainsi que la sélection et le choix des clients qui vont participer à chaque tour du processus d'apprentissage [53].

II.3.4.1 Problèmes liés à la sélection des clients

Nous citons les différents problèmes liés à la sélection des clients :

— **Hétérogénéité des données** : les données des différents clients peuvent varier en termes de caractéristiques, de distributions, de qualité et de quantité. Par exemple, un client peut avoir des données provenant d'un ensemble de capteurs différents de ceux des autres clients, ce qui veut dire des différences dans la qualité des données, ce qui rend difficile la comparaison des caractéristiques des données [29].

— **Communication des modèles entre les clients et le serveur** : ce problème se pose lorsque le nombre de clients est élevé, car la transmission de leurs mises à jour au serveur peut être coûteuse en termes de temps et de bande passante. De plus, des problèmes de latence et de stabilité de la connexion peuvent survenir, ce qui peut retarder ou interrompre la transmission des mises à jour.

— **Biais dans la sélection des clients** : les méthodes de sélection des clients peuvent introduire un biais dans les données utilisées pour l'entraînement du modèle, ce qui peut affecter la qualité des prédictions [54].

— **Clients aux ressources limitées** : l'apprentissage fédéré nécessite une grande quantité de ressources de calcul et de communication, en particulier pour les clients ayant des données de grande taille, ce qui peut limiter la participation des clients à capacités

réduites, et ainsi, des données contenant des informations pertinentes peuvent être ignorées [55].

II.3.4.2 Stratégies de sélection des clients

Afin de résoudre le problème de sélection de clients dans l'apprentissage fédéré, plusieurs méthodes ont été proposées dans la littérature. Parmi lesquelles, nous citons :

— **Sélection partielle des clients** : cette approche consiste à sélectionner un sous-ensemble aléatoire de clients pour participer au processus d'apprentissage fédéré, sans prendre en compte les spécifications des clients ou leur contribution au calcul du modèle global. Cette méthode est efficace pour les ensembles de données de grande taille et hétérogènes, mais elle n'est pas optimale pour des ensembles de données plus petits ou dans les cas où les clients ont des capacités différentes. De plus, si les clients les plus performants ne sont pas inclus dans le sous-ensemble sélectionné, cela peut entraîner une convergence plus lente ou une qualité de modèle moins bonne [55].

— **Analyse de convergence** : cette approche consiste à prendre en compte la fonction de perte de chaque client pour sélectionner les clients ayant une fonction de perte minimale. Cela garantit que les clients qui contribuent le plus à l'agrégation sont sélectionnés. Cette approche est souvent utilisée pour des ensembles de données plus petits et homogènes où les clients ont des capacités relativement similaires [56].

— **Mise à jour par rapport au modèle local** : cette approche consiste à évaluer la contribution de chaque client en utilisant la norme de la mise à jour par rapport au modèle local. Cette contribution est ensuite transformée en une probabilité qui permet de sélectionner les clients les plus susceptibles de participer à l'entraînement. Cette approche prend en compte les différences entre les clients en termes de performance et de quantité de données. Cela permet une sélection plus équitable et peut améliorer la qualité du modèle global [57].

— **Groupement et sélection fédérés (FedCS)** : cette approche a été développée pour adapter le processus d'apprentissage en fonction des capacités des clients. Les clients avec des ressources limitées ne contribuent pas au modèle global, ce qui permet une convergence plus rapide par rapport à une sélection aléatoire. Cela peut être particulièrement utile dans les cas où les clients ont des capacités de traitement différentes ou lorsqu'ils utilisent des appareils mobiles avec des ressources limitées. En utilisant cette approche, les ressources des clients sont mieux utilisées pour améliorer la qualité du modèle global [58].

— **Apprentissage fédéré favorisant la diversité (DivFL)** : c'est une technique utilisée pour sélectionner un petit sous-ensemble de clients pour communiquer leurs mises à jour de modèle au serveur à chaque tour d'entraînement.

L'objectif de la sélection de clients diversifiés est de choisir des clients représentatifs à partir de l'ensemble des données, en termes de distribution de données et de diversité des appareils clients. Cela est important, car si le serveur ne reçoit des mises à jour que d'un petit nombre de clients, le modèle pourrait ne pas être capable d'apprendre de la pleine diversité des données [59].

II.4 Taxonomies des systèmes d'apprentissage fédéré

La diversité des systèmes d'apprentissage fédéré rend difficile la compréhension de leurs caractéristiques et de leur utilisation. Pour y remédier, nous avons établi une taxonomie des systèmes d'apprentissage englobant ses différents aspects, tels que présentée dans la Fig. II.2.

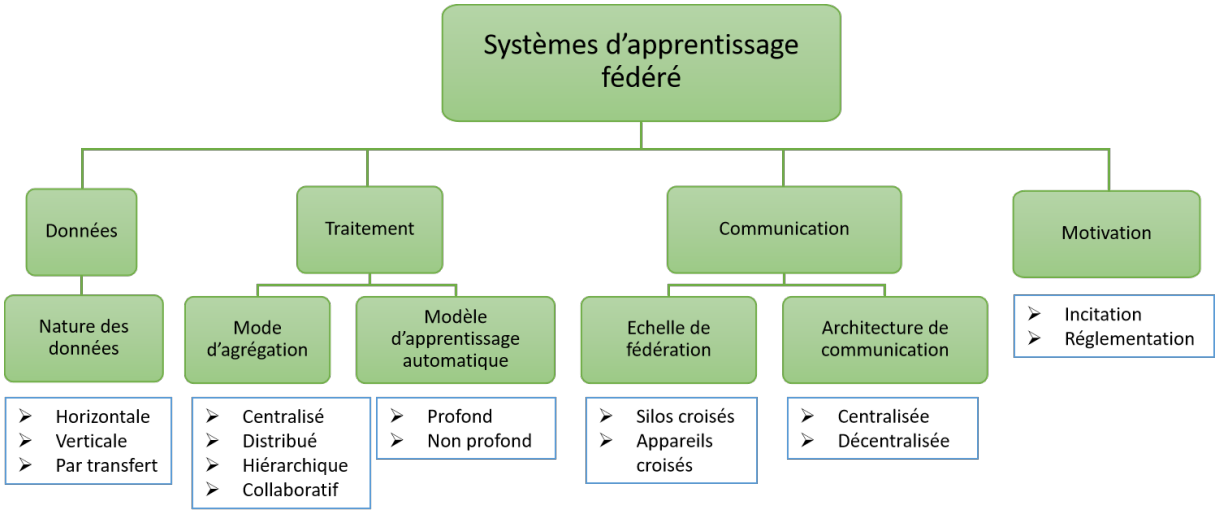


Figure II.2 — Taxonomie des systèmes d'apprentissage fédéré, inspiré de [60].

Nous détaillons dans ce qui suit les familles présentées dans la taxonomie.

a. **Nature de données** : il existe trois types de données utilisées dans les systèmes fédérés : les données horizontales, les données verticales et les données de transfert [61].

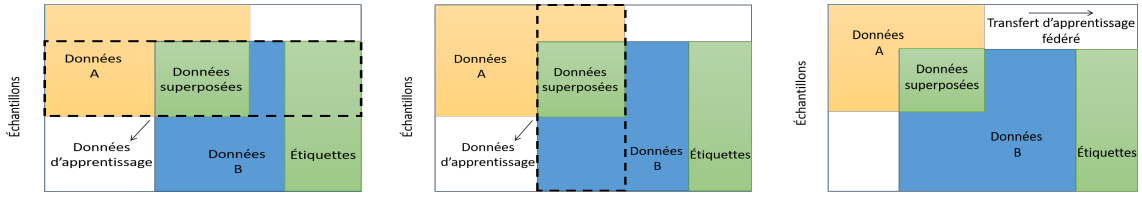
b. **Type de traitement** : il existe deux types de traitement appliqués dans les systèmes fédérés : le traitement selon la catégorie du modèle d'apprentissage et le traitement selon le mode d'agrégation.

c. **Type de communication** : il existe deux aspects importants de la communication dans les systèmes fédérés : l'échelle de fédération et l'architecture de communication.

d. **Motivation** : il existe deux types de motivations derrière le développement des systèmes fédérés : l'incitation et la réglementation.

II.4.1 Catégorisation selon la nature des données

L'application de l'apprentissage fédéré peut différer selon la nature des données utilisées par chacun des clients [61]. Nous distinguons trois catégories qui sont montrées dans la Fig. II.3.



(a) Données horizontales (b) Données verticales (c) Données par transfert

Figure II.3 — Répartition des données dans l'apprentissage fédéré [61].

a. **Apprentissage fédéré horizontal** : il est connu aussi sous le nom d'apprentissage fédéré homogène. Dans ce cas, les données que possèdent les différents clients partagent le même espace de caractéristiques. Autrement dit, chaque record ou donnée est défini par un vecteur du même format $(x_1, x_2, \ldots, x_n; y)$ où les x_i représentent les caractéristiques et y représente la classe de la donnée. Ce type d'apprentissage fédéré est le plus couramment utilisé [61]. La reconnaissance des mots d'éveil (*Wake-word recognition*) [62], tels que " Hey Siri " et " OK Google ", est une application typique de la répartition horizontale, car chaque utilisateur prononce la même phrase avec une voix différente.

b. **Apprentissage fédéré vertical** : dans ce type d'apprentissage, la répartition des données n'est pas faite à la base d'échantillons, mais à la base des caractéristiques. Autrement dit, les différents clients possèdent le même échantillon de données, mais pas la totalité des caractéristiques.

   Les techniques d'apprentissage vertical utilisent une architecture d'entraînement différente de celle des données partitionnées horizontalement. En fonction de l'algorithme d'agrégation, les clients échangent des résultats intermédiaires spécifiques plutôt que des paramètres de modèle, afin d'aider les calculs de gradient des autres clients [63].

   Pour bien illustrer l'apprentissage vertical, nous donnons l'exemple d'utilisation de l'apprentissage vertical (VFL) dans le contexte d'une compagnie d'assurance automobile qui souhaite améliorer son modèle d'évaluation des risques en incorporant des attributs supplémentaires provenant d'autres entreprises, telles qu'une banque ou un bureau des impôts. Dans ce cas, les participants collaborerent pour former un modèle en utilisant des données verticalement partitionnées, où chaque participant

possède les étiquettes de ses propres attributs [63].

Ce type d'apprentissage permet d'obtenir des informations supplémentaires sans partager directement les données entre les participants, tout en nécessitant moins de ressources que le partage de données direct. Cela rend l'entraînement distribué plus léger et variable à l'échelle.

c. **Apprentissage fédéré par transfert** : il s'agit d'un cas particulier d'apprentissage fédéré et il diffère à la fois de l'apprentissage fédéré horizontal et vertical. L'idée est d'entraîner un modèle en utilisant nos connaissances, mais à la base d'un modèle issu d'une autre application. La finalité derrière cette configuration est d'orienter le modèle de base, pré-entraîné sur des ensembles de données généralisées, pour une application spécifique en entraînant une faible quantité de données. Il vise donc à construire un modèle efficace pour un domaine cible tout en exploitant les connaissances des autres domaines (sources) [64].

II.4.2 Catégorisation selon le type de traitement

Les systèmes d'apprentissage fédéré procèdent à une agrégation des modèles locaux issus des différents clients participants dans le processus d'entraînement pour aboutir à un modèle global, censé être performant, cohérent et efficace. Le mode d'agrégation est utilisé pour déterminer comment les modèles sont combinées et utilisées dans le processus d'apprentissage.

II.4.2.1 Modes d'agrégation

La dernière étape dans le processus d'apprentissage fédéré vise à l'agrégation des différents modèles locaux établis par chacun des clients. Cette agrégation peut s'effectuer selon quatre (04) modes différents qui sont présentés dans la Fig II.4, à savoir l'agrégation centralisée, l'agrégation distribuée, l'agrégation hiérarchique et l'agrégation collaborative [9].

a. **Agrégation centralisée** : un seul serveur périphérique se charge de l'agrégation des modèles d'apprentissage locaux de tous les dispositifs, c'est-à-dire, l'agrégation se fait à un seul (01) niveau. La Fig. II.4(a) montre ce mode d'agrégation qui se fait en trois étapes.

Dans un premier temps, les clients calculent leurs modèles locaux (étape 1). Ils les partagent par la suite avec le serveur (étape 2), qui calcule le modèle global en agrégeant les modèles locaux qu'il reçoit (étape 3).

b. **Agrégation distribuée** : contrairement à l'agrégation centralisée, ce mode implique plusieurs serveurs d'agrégation qui reçoivent les mises à jour des modèles d'ap-

prentissage locaux de leurs dispositifs associés. La Fig. II.4(b) montre l'agrégation distribuée qui se fait en quatre (04) étapes.

L'agrégation se fait en deux (02) étapes. Tout d'abord, chaque groupement de clients/serveur procède à une agrégation centralisée. Puis, les modèles globaux des différents serveurs sont partagés entre eux pour une agrégation plus globale (étape 4).

c. **Agrégation hiérarchique** : l'agrégation hiérarchique consiste à appliquer l'agrégation à plusieurs niveaux de serveurs. Autrement dit, l'architecture de communication est vu comme un arbre auquel les feuilles sont les clients et les serveurs d'agrégation sont les nœuds internes. Le serveur central est la racine de l'arbre. Par exemple, il peut y avoir des serveurs de périphériques ou bien des serveurs par type de dispositifs avant d'opérer une agrégation qui englobe tous les modèles locaux. Cela peut entraîner une augmentation de la latence lors de la transmission de modèles locaux à travers plusieurs niveaux de serveurs avant d'atteindre le serveur central pour l'agrégation globale. Ce mode est illustré dans la Fig. II.4(c), où une cinquième étape est ajoutée.

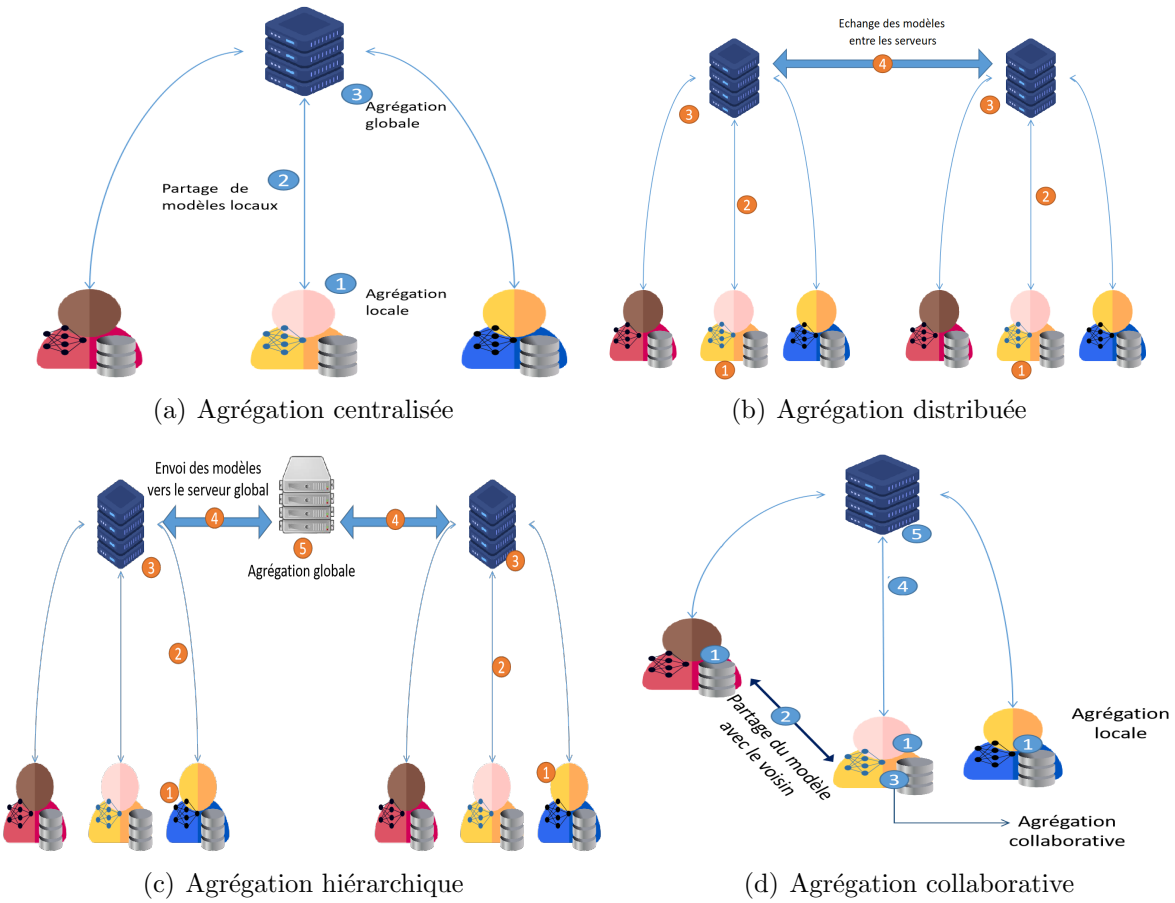
d. **Agrégation collaborative** : dans tous les modes décrits précédemment, les clients sont supposés avoir assez de ressources de calcul et de communication pour pouvoir participer au processus d'apprentissage. Pour remédier au problème engendré dans le cas contraire, l'apprentissage fédéré collaboratif est apparu. Il a été introduit principalement pour assurer la participation des dispositifs qui ne peuvent pas communiquer avec le serveur central [65].

   Dans un tel scénario, les clients à capacités réduites envoient leurs modèles à d'autres clients proches disposant de ressources de communication suffisantes. Les dispositifs récepteurs effectuent l'agrégation des modèles locaux des autres dispositifs avec leurs propres modèles et envoient par la suite les paramètres du modèle agrégé au serveur central. Un schéma illustrant ce processus est donné dans la Fig. II.4(d).

   Le client en rouge (premier client de gauche) calcule son modèle local (étape 1), mais il n'a pas assez de ressources de communication pour le partager avec le serveur. Du coup, il le partage avec son voisin (étape 2). Ce dernier calcule l'agrégat des deux modèles (étape 3). Il envoie, par la suite, ainsi que tous les autres clients, le modèle résultant au serveur (étape 4), qui calcule le modèle global (étape 5).

II.4.2.2 Modèles d'apprentissage machine

En utilisant l'apprentissage fédéré pour résoudre des problèmes d'apprentissage automatique, les clients souhaitent généralement entraîner un modèle d'apprentissage automatique de dernière génération sur une tâche spécifiée. Il y a eu de nombreux efforts pour



(a) Agrégation centralisée (b) Agrégation distribuée

(c) Agrégation hiérarchique (d) Agrégation collaborative

Figure II.4 — Modes d'agrégation dans l'apprentissage fédéré [9].

développer de nouveaux modèles ou réinventer les modèles actuels dans un environnement fédéré. Ils peuvent être divisés en deux (02) catégories : profonds et non profonds.

a. **Modèles non profonds** : ce sont des modèles d'apprentissage automatique qui ont une architecture avec un nombre limité de couches cachées. Citons les modèles linéaires qui sont classiques et faciles à utiliser, par exemple la régression linéaire. Il existe des systèmes bien développés pour la régression linéaire [66] et la régression logistique [35]. Ces modèles linéaires sont faciles à entraîner par rapport aux autres modèles complexes tels que les réseaux de neurones.

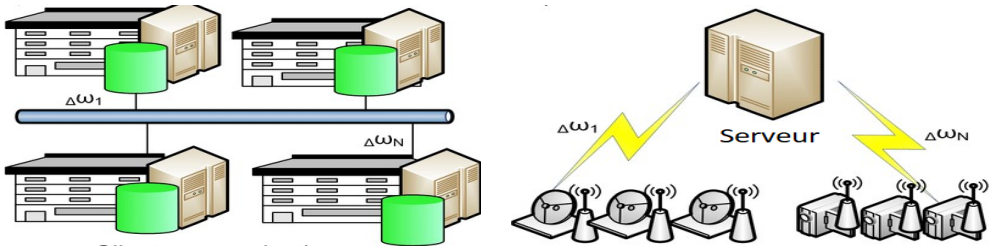
b. **Modèles profonds** : les modèles profonds sont des types de modèles d'apprentissage automatique qui utilisent plusieurs couches de neurones artificiels. Chaque couche effectue des transformations sur les données d'entrée afin de produire des résultats pour des besoins spécifiques. Ces modèles ont été utilisés pour obtenir des résultats de pointe dans plusieurs tâches, telles que la classification d'images et la prédiction de mots [67, 68].

II.4.3 Catégorisation selon le mode de communication

L'échelle de fédération et l'architecture de communication font référence à la manière dont les différents systèmes d'apprentissage fédéré sont organisés et interconnectés pour permettre une communication efficace et fluide entre les différents niveaux et parties prenantes d'une organisation. L'échelle de fédération décrit la manière dont les différents systèmes sont hiérarchisés et organisés en fonction de leur importance et de leur rôle dans l'organisation. L'architecture de communication décrit les différents moyens et protocoles utilisés pour faciliter la communication entre les différents systèmes de l'échelle de fédération.

II.4.3.1 Échelles de la fédération

Les systèmes d'apprentissage fédéré peuvent être catégorisés en deux types en fonction de l'échelle de la fédération : les systèmes basés sur les silos croisés et les systèmes basés sur les appareils croisés. Les différences entre eux réside dans le nombre de clients et la quantité de données stockées dans chaque client, ce qui peut affecter grandement les performances du système d'apprentissage fédéré [32]. La Fig. II.5 montre les deux types de l'échelle de fédération.



(a) Silos croisés (b) Appareils croisés

Figure II.5 — Échelles de fédération dans l'apprentissage fédéré [69].

a. **Silos croisés (*Cross-silos*)** : il permet aux organisations (par exemple, financières ou médicales) de former collaborativement un modèle d'apprentissage automatique en agrégeant les mises à jour de gradient locales de chaque client sans partager de données sensibles à la vie privée, tel que présenté dans la Fig. II.5(a). Il y a généralement un nombre relativement faible de clients et chacun d'entre eux possède une quantité relativement importante de données ainsi que des ressources de calcul. Durrant et al. [70] expliquent comment l'apprentissage fédéré peut résoudre le problème de la protection des données dans le secteur agroalimentaire, où les données sont souvent considérées comme des actifs précieux. Les auteurs proposent une solution technique basée sur l'apprentissage fédéré qui utilise des données décentralisées

pour développer un modèle d'apprentissage machine croisé qui facilite le partage de données entre les chaînes d'approvisionnement. Ils se concentrent sur l'amélioration de l'optimisation de la production par la prédiction du rendement du soja, et fournissent des cas d'utilisation potentiels pour aider à résoudre d'autres problèmes. Les résultats montrent que leur approche est plus performante que chaque modèle entraîné individuellement sur une source de données, et que le partage de données dans le secteur agroalimentaire peut être facilité grâce à des alternatives à l'échange de données, tout en contribuant à l'adoption de technologies d'apprentissage machine émergentes pour augmenter la productivité.

b. **Appareils croisés (*Cross-devices*)** : dans ce type de systèmes, illustré dans la Fig. II.5(b), le nombre de clients est généralement élevé. Chaque client possède une quantité relativement faible de données ainsi que des ressources de calcul, ils s'agissent généralement des appareils mobiles [71]. Google Keyboard [72] est un exemple de systèmes d'apprentissage fédéré basé sur les appareils croisés. Les suggestions de ses requêtes peuvent être améliorées à l'aide de l'apprentissage fédéré. En raison de la préoccupation pour la consommation d'énergie, les clients ne peuvent pas être invités à effectuer des tâches d'entraînement complexes. Dans ce cas, le système doit être suffisamment puissant pour gérer un grand nombre de clients et faire face aux problèmes possibles tels que la connexion instable entre ces clients et le serveur.

II.4.3.2 Architectures de communication

L'architecture des systèmes d'apprentissage fédéré peut varier en fonction du type de communication entre ses différents acteurs. Nous citons les deux types de conception des architectures de communication les plus utilisées, à savoir la communication centralisée et la communication décentralisée.

a. **Architecture centralisée** : le flux de données est souvent asymétrique, ce que signifie que le gestionnaire agrège les informations (par exemple, les modèles locaux) des clients et renvoie les résultats [73]. Les mises à jour des paramètres sur le modèle global sont toujours effectuées par ce gestionnaire. La communication entre le gestionnaire et les clients locaux peut être synchrone [55] ou asynchrone [74]. La scalabilité et la stabilité (tolérance aux pannes) sont deux facteurs importants dans ce type de conception. Parmi les architectures centralisées, nous citons l'architecture client/serveur, n-tiers, etc.

b. **Architecture décentralisée** : dans ce genre d'architecture, toute dépendance à

l'égard du serveur central pour l'agrégation des modèles est exclue. L'entité centrale est substituée par des algorithmes pour établir la confiance et la fiabilité. Il n'existe pas d'entité centrale qui calcule le modèle global. Au lieu de cela, chaque participant calcule son propre modèle local et le partage avec ses voisins pour élaborer le modèle global. Ainsi, des algorithmes distribués sont utilisés pour la diffusion et le partage des modèles entre les différentes entités du système fédéré [75].

Parmi les architectures décentralisées, nous citons l'architecture pair-à-pair (*Peer to peer*, P2P), les réseaux blockchain, les réseaux de capteurs sans fil, etc.

Lalitha et al. [76] ont utilisé une architecture P2P pour le processus d'apprentissage fédéré sur des graphes. Pour cela, les nœuds ont employé une approche bayésienne en ajoutant une croyance sur l'espace des paramètres du modèle. Ils ont développé un algorithme d'apprentissage distribué dans lequel les nœuds ont mis à jour leur croyance en agrégeant les informations de leurs données locales avec le modèle de leurs voisins à une distance d'un saut. Cette méthode a permis d'apprendre de manière collective un modèle qui correspondait le mieux aux observations sur l'ensemble du réseau. Li et al. [77] présentent un *framework* d'apprentissage fédéré basé sur la blockchain, nommé BFLC, qui définit les modèles de stockage, le processus d'entraînement et un nouveau consensus de comité en détail.

Bien que l'architecture centralisée soit largement utilisée dans les études existantes, la conception décentralisée est préférée pour certains aspects, tels qu'une qu'une amélioration de la confidentialité et de la sécurité, ainsi qu'une réduction de la latence. Ceci est dû au fait que la concentration des informations sur un serveur peut présenter des risques, notamment le problème du point de défaillance unique ({*Single Point Of Failure* (SPOF)). La principale cause de ce problème est la dépendance à un serveur central pour coordonner le processus d'apprentissage. En contrepartie, dans l'architecture décentralisée, un mécanisme d'élection d'un serveur central est effectué. Ce serveur élu est responsable de l'agrégation des mises à jour des dispositifs individuels, de la mise à jour du modèle global et de la distribution du modèle mis à jour aux dispositifs. Si ce serveur échoue, l'ensemble du processus d'apprentissage sera interrompu [78]. Pour y remédier, le protocole de tolérance aux fautes byzantine est un bon exemple pour garantir que le système peut continuer à fonctionner correctement même en présence de défaillances ou de comportements malveillants [79].

II.4.4 Catégorisation selon la motivation

Dans le monde réel, plusieurs entités ont besoin de motivation pour qu'elles adhèrent à l'idée d'adopter un système d'apprentissage fédéré. Cette motivation peut être due soit

à des réglementations ou bien à des incitations.

a. **Réglementation** : l'utilisation de l'apprentissage fédéré au sein d'une entreprise ou d'une organisation est souvent motivée par des réglementations. Par exemple, un département qui possède des enregistrements de transactions des utilisateurs peut aider un autre département à prédire la solvabilité d'un utilisateur en utilisant l'apprentissage fédéré. Dans de nombreux cas, les clients ne peuvent pas être obligés de fournir leurs données en raison des réglementations en vigueur. Cependant, les clients qui choisissent de participer à l'apprentissage fédéré peuvent en bénéficier, comme une meilleure précision du modèle. Un exemple de ce type de participation est le clavier Google [72]. Bien que les utilisateurs aient la possibilité d'empêcher le clavier Google d'utiliser leurs données, ceux qui acceptent de télécharger les données d'entrée peuvent bénéficier d'une prédiction de mots plus précise. Les utilisateurs peuvent être enclins à participer à l'apprentissage fédéré pour leur propre confort.

b. **Incitation** : ce type de système se base principalement sur l'incitation des utilisateurs (clients) à participer à l'alimentation du système en données, en garantissant au client qui fournit le plus de données à tirer un plus grand avantage de l'apprentissage fédéré. Des cas réussis de conception d'incitations dans la blockchain ont été signalés [80, 81]. Les clients à l'intérieur du système peuvent être des collaborateurs ainsi que des concurrents. D'autres conceptions d'incitation sont proposées pour attirer les participants avec des données de haute qualité pour l'apprentissage fédéré [82, 83].

Algorithmes d'agrégation en apprentissage fédéré

La phase d'agrégation consiste à regrouper tous les modèles locaux issus des différents clients et appliquer une opération d'agrégation pour aboutir à un modèle global. Les algorithmes d'agrégation applicables dans l'apprentissage fédéré sont nombreux. L'algorithme de base est celui proposé par Google en réponse au problème de prédiction du prochain mot dans les tâches de saisie par clavier. Dans cette section, nous allons expliquer cet algorithme et son principe de fonctionnement ainsi que citer d'autres algorithmes d'agrégation.

Description de l'algorithme FedAvg

Le Federated Averaging (FedAvg), également connu sous le nom d'Algorithme 1, est le premier algorithme d'agrégation de modèles locaux dans un processus d'apprentissage fédéré. Il a été introduit par McMahan et al [55]. Le principe de cet algorithme est d'ef-

fectuer un moyennage sur l'ensemble des modèles locaux issus des clients. Il est contrôlé par les trois paramètres suivants :

— La fraction de clients participants dans le processus d'apprentissage, notée C.

— Le nombre d'époques qu'exécute chaque client localement, noté E.

— La taille du lot de données utilisées pour l'entraînement local, notée B.

Algorithme 1 Moyennage fédéré (Federated Averaging) (Exécution au niveau du serveur)

initialiser ω_0. ▷ ω_0 est le modèle global stocké au niveau du serveur.
for $t = 1 \dots I$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (un ensemble aléatoire de m clients)
    for all client $k \in S_t$ en parallèle **do**
        $\omega_{k,t+1} \leftarrow \text{MiseAJourClient}(k, \omega_t)$
    end for
    $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{k,t+1}$
    envoyer ω_{t+1} à tous les K clients.
end for

D'autres paramètres constants entrent en jeu pour la configuration du processus, à savoir le nombre de clients connectés au serveur, noté K, et le taux d'apprentissage η. P_k est le jeu de données du client k et n_k est le nombre de données que le client k possède ($n_k = |P_k|$), n est le nombre total de données mises en jeu pour l'apprentissage et I représente le nombre total d'itérations fixé par le serveur.

Quand un processus d'apprentissage fédéré est lancé, le serveur prépare l'ancien modèle global s'il existe, ainsi que les paramètres d'apprentissage et les envoie aux différents clients (dans le cas d'un entraînement basé sur un modèle pré-entraîné).

À chaque itération t, le serveur construit un ensemble S_t constitué de m clients sélectionnés aléatoirement. Ensuite, chaque client procède à la mise à jour de son modèle local à travers la descente stochastique du gradient (Stochastic Gradient Descent (SGD)). Pour ce faire, il exécute l'Algorithme 2.

Algorithme 2 Mise à jour client (k, ω) (Exécution au niveau du client k)

$B \leftarrow$ (diviser P_k en lots de taille B)
for epoch $= 1 \dots E$ **do**
    for batch $b \in B$ **do**
        $\omega \leftarrow \omega - \eta \nabla l(\omega; b)$
    end for
end for
return ω

Chaque client commence d'abord par diviser son jeu de données local en un ensemble de lots de taille B définie par le serveur. Pour chaque époque, il exécute la propagation

autant de fois qu'il y a de lots. À la fin, il émet son modèle local calculé. Le calcul des modèles locaux de tous les clients se fait en parallèle pour gagner du temps. Une fois un client ayant terminé le calcul, il envoie son modèle au serveur. Quand le serveur reçoit tous les modèles locaux, il procède à une opération de moyennage en tenant compte de l'hétérogénéité statistique. Chaque modèle local est pondéré par la proportion de données que le client propriétaire possède.

Autres techniques d'optimisation et d'agrégation des modèles

En plus de FedAvg, plusieurs algorithmes et techniques d'agrégation ont été introduits, à savoir FedProx [84], FedDANE [85] et autres. Nous décrivons dans cette section quelques techniques qui ont impliqué de nouveaux concepts pour répondre au problème d'optimisation et d'agrégation des modèles.

FedProx

Pour répondre au problème engendré par la différence en termes de ressources des différents clients, Li et al. [84] ont proposé un nouvel algorithme appelé FedProx, similaire à FedAvg. Cet algorithme suit les mêmes étapes que FedAvg concernant la sélection des clients et la façon d'agrégation des modèles. Cependant, il introduit une régularisation dans la fonction de perte en ajoutant un terme proximal qui vise à limiter la quantité de travail effectuée par le client, et de ce fait, la perte sera comme suit :

$$\min_{w} h_k(w; w^t) = F_k(w) + \underbrace{\frac{\mu}{2}\|w - w^t\|^2}_{\text{terme proximal}} \tag{II.1}$$

Le terme proximal introduit dans l'équation (II.1) mesure la distance entre le modèle local calculé par le client et le modèle global agrégé par le serveur. La distance est pondérée par un coefficient μ qui représente un hyper-paramètre pour un réglage adaptatif, introduit pour cerner l'impact de la distance sur la fonction de perte. Selon leurs résultats [84], le terme proximal dans l'équation (II.1) limite efficacement l'impact des mises à jour locales (en les restreignant à la proximité du point initial) sans qu'il soit nécessaire d'ajuster manuellement le nombre local d'époques. Il est à noter que lorsque $\mu = 0$, on revient à l'optimisation effectuée par FedAvg.

SCAFFOLD

Il s'agit d'un autre algorithme similaire à FedAvg, proposé par Karimireddy et al. [86]. Les auteurs estiment que l'hétérogénéité des clients a souvent un impact négatif sur FedAvg. Pour y remédier, SCAFFOLD (Stochastic Controlled Averaging for Federated Learning) introduit un terme de correction pour éviter la dérive des clients (client-drift) lors de

l'agrégation. La dérive des clients est définie comme la tendance de chaque client à pousser le modèle global dans une direction différente dans l'espace d'optimisation, éloignant ainsi le modèle de sa version optimale [87]. Ce problème a été observé dans les processus d'apprentissage fédéré continu, où l'apprentissage des clients se fait de manière séquentielle et l'agrégation est réalisée de façon incrémentale. Comme solution, SCAFFOLD tente de corriger cette dérive en estimant la direction de mise à jour pour le modèle du serveur c et la direction de mise à jour pour chaque client c_i. La différence $c - c_i$ est alors utilisée comme estimation de la dérive du client pour corriger la mise à jour locale.

FedDANE

Un défi critique dans les approches d'apprentissage fédéré réside dans le coût de communication et d'échange des messages entre les différents acteurs du système fédéré. Le calcul du modèle global nécessite que le serveur communique avec tous les clients et fasse ensuite la moyenne des modèles locaux, ce qui est infaisable dans les réseaux massifs avec un grand nombre de clients.

Pour répondre à ce problème, Tian et al. [88] ont proposé une approche basée sur la technique d'optimisation distribuée DANE, et l'ont nommée FedDANE. Ils suggèrent d'approximer l'ensemble des modèles locaux reçus par le serveur à un sous-ensemble de modèles provenant de clients échantillonnés de manière aléatoire. Ainsi, l'agrégation se décompose en deux phases. Dans un premier temps, le serveur agrège les modèles des clients sélectionnés aléatoirement (comme dans FedAvg). Puis, il envoie le modèle global calculé à un autre groupe de clients qui, chacun, calcule son propre modèle local et l'envoie au serveur. Ce dernier applique alors une deuxième phase d'agrégation sur les modèles de ces clients.

II.5 Applications de l'apprentissage fédéré dans le domaine médical

Les applications de l'apprentissage fédéré s'étendent sur un certain nombre d'industries notamment les télécommunications [89], la cybersécurité [90], le traitement du langage naturel [91], l'analyse financière [92] et la médecine [29]. Nous nous concentrons dans cette section à présenter quelques travaux connexes relatifs à l'application de l'apprentissage fédéré dans le domaine médical.

Yuan et al. proposent un nouveau framework pour l'application de l'apprentissage fédéré dans un contexte IoT à base des réseaux fractionnés. Cette réflexion repose sur l'hypothèse que le temps d'apprentissage va décroître considérablement sous prétexte que l'énergie des dispositifs connectés soit un critère crucial pour le bon fonctionnement du système fédéré. Ils procèdent à une décomposition du réseau de neurones en deux sous-réseaux. Le premier réseau est chargé du traitement des données du client mis au niveau

du dispositif IoT. Quant au deuxième, il est caractérisé par sa large profondeur. Il est mis dans le cloud et se charge de l'agrégation et la mise à jour du modèle global [93].

Pour les dispositifs médicaux portables, les données des clients sont souvent isolées et du coup l'agrégation peut ne pas s'achever sans compromettre leur confidentialité. D'une autre part, les modèles agrégés échouent en matière de personnalisation en raison de l'empoisonnement. Pour remédier à ces deux problèmes, Chen et al. introduisent le framework FedHealth qui est le premier qui se base sur le transfert fédéré pour le diagnostic auxiliaire de la maladie de Parkinson à partir des images CT collectées depuis les dispositifs de l'imagerie médicale (Scanner et IRM) [94].

Pour la détection du cancer du sein, Jimenez et al. ont développé un système collaboratif de diagnostic assisté par ordinateur. Ce système qui se base sur l'apprentissage fédéré ne repose pas sur des données de la même modalité, mais qui fait intervenir des clients avec des données hétérogènes et qui ne sont pas identiquement distribuées (non-iid heterogeneous data) [**?**].

Cetinkaya et al. ont appliqué l'apprentissage fédéré pour la détection de la pneumonie et COVID-19, mais dans un environnement cloud. Ce choix est justifié par le fait de vouloir réduire la communication entre le serveur et les clients en minimisant l'échange de données entre eux. Plus encore, ils ont introduit le concept de clustered federated learning. Cela revient à entraîner les modèles sur des images de différentes modalités (X-Ray et Ultrasound) avec de mêmes étiquettes. Autrement dit, l'espace de caractéristiques n'est pas le même contrairement à l'espace de classification. Les clients sont subdivisés ainsi en deux clusters. Le premier cluster regroupe les clients possédant les images X-Ray et le deuxième ne détient que les images Ultrasound. Chaque groupe entraîne son modèle avec ses propres données. Le serveur n'a ensuite qu'à agréger les modèles locaux calculés. Cette configuration a montré une amélioration de 16

D'un autre côté, grâce aux récents développements de l'intelligence artificielle, les applications de la télémédecine ont suscité un grand intérêt dans le monde entier. Cependant, les approches existantes pour les tâches de soins à domicile ne prêtent pas suffisamment attention à la confidentialité des données des utilisateurs. Elles sont donc loin d'être prêtes pour un déploiement pratique à grande échelle. Pour remédier à ce problème, Qiong et al. ont proposé un nouveau framework d'apprentissage fédéré basé cloud pour le suivi des patients à domicile, appelé FedHome. Les auteurs ont proposé un système constitué d'un serveur cloud et N nœuds déployés au domicile des utilisateurs. Dans le processus d'entraînement de FedHome, le serveur cloud initialise d'abord le modèle, puis l'envoie à tous les clients participants (par exemple, les smartphones). Ensuite, chaque utilisateur de dispositif IoT peut soit effectuer la tâche d'entraînement chez lui ou la déléguer à un

nœud fiable (par exemple, passerelle intelligente) dans la maison pour un calcul rapide. Enfin, un modèle global peut être bien agrégé en tirant parti des mises à jour des modèles de plusieurs terminaux domestiques sous la coordination du serveur de cloud central [**?**].

Discussion et synthèse

Le parcours sur les travaux connexes concernant l'application de l'apprentissage fédéré dans le domaine médical nous permet de relever le manque de l'aspect réaliste. Ceci nous pousse à mettre en œuvre une plateforme générique permettant la collaboration de différents clients, chacun dans son domaine, pour établir des modèles intelligents aidant le bon diagnostic des différentes maladies et la bonne gestion des records hospitaliers des différents patients.

Un système intelligent qui combine les différents types de capteurs et appareils médicaux est devenu une nécessité vu que la santé des individus est d'une priorité primordiale. C'est dans ce sens que nous avons choisi d'appliquer les techniques d'apprentissage fédéré sur des données issues des différents capteurs et appareils médicaux, ce qui est relativement non pris en compte dans la plupart des travaux relevant le problème de confidentialité.

Les travaux cités et d'autres se sont concentrés le plus sur l'application de l'apprentissage fédéré sans prendre en considération l'aspect d'amélioration des modèles intelligents du point de vue des défis relevés précédemment, notamment la capacité des clients à entraîner les modèles.

II.6 Conclusion

Avec l'évolution rapide des technologies de traitement et d'analyse, et vu le gros volume de données qui circulent entre des milliers de dispositifs, les approches d'apprentissage profond sont devenues d'une nécessité primordiale pour minimiser l'effort de traitement et le temps de calcul. De plus, en raison du privilège de la confidentialité et la sécurité de la vie privée des utilisateurs, le concept de l'apprentissage fédéré a été introduit. L'apprentissage fédéré est utilisé principalement pour renforcer la confidentialité des données en décentralisant la phase d'entraînement pour qu'elle soit effectuée sur chacun des clients détenant son ensemble de données sans qu'il le partage.

Dans ce chapitre, nous avons approfondi l'explication de l'apprentissage fédéré, son processus et ses défis. Nous avons réalisé une taxonomie résumant l'ensemble des systèmes d'apprentissage fédéré ainsi que quelques travaux qui mettent en œuvre l'apprentissage fédéré dans le domaine médical. Vu que notre projet vise à étudier le problème de reconnaissance de pathologies à base d'apprentissage fédéré, le prochain chapitre sera dédié à

l'evaluation de cet apprentissage dans un cas précis.

Chapitre III

EVALUATION DES APPROCHES D'APPRENTISSAGE FEDERE POUR LA DETECTION DE PNEUMONIE

III.1 Introduction

L'évolution rapide des technologies de traitement des données transforme le domaine de la santé. La détection précoce des maladies, notamment la pneumonie, est essentielle pour améliorer les résultats des patients. Traditionnellement, le diagnostic de la pneumonie repose sur des examens cliniques et radiologiques, nécessitant une expertise considérable.

L'apprentissage profond, et en particulier les réseaux de neurones convolutifs (CNN), améliore l'automatisation et la précision des diagnostics à partir d'images médicales. Cependant, l'entraînement de ces modèles requiert de grandes quantités de données sensibles, posant des défis en matière de confidentialité.

L'apprentissage fédéré offre une solution innovante en permettant de former des modèles sur des données décentralisées, protégeant ainsi la confidentialité des données des patients et facilitant la collaboration entre institutions médicales.

Ce chapitre explore l'utilisation de l'apprentissage fédéré pour détecter la pneumonie à partir d'images de radiographies pulmonaires. Nous décrivons la conception de la recherche, les méthodes de collecte et de prétraitement des données, les réseaux de neurones convolutifs et la comparaison des architectures de modèles VGG16, ResNet50 et InceptionV3.

Nous abordons également l'apprentissage fédéré, expliquant la formation décentralisée des modèles et l'agrégation des mises à jour. Enfin, nous évaluons les performances des modèles à l'aide de métriques standardisées.

III.2 Méthodologie de travail

L'application des techniques d'apprentissage profond nécessite une méthodologie de travail bien spécifique qui doit être conforme au processus d'apprentissage présenté dans la Fig. III.1

Dans ce sens, nous avons divisé le travail en quatre (04) principales étapes, à savoir la préparation des données, la définition de l'architecture utilisée, l'entraînement du modèle et enfin la mesure de son exactitude.

III.2.1 Préparation des données

Pour la détection de la pneumonie, nous avons choisi de travailler avec des jeux de données contenant des images radiographiques. Nous avons sélectionné le jeu de données du Guangzhou Women and Children's Medical Center (GWCMC), collecté par Kermani et al [95].

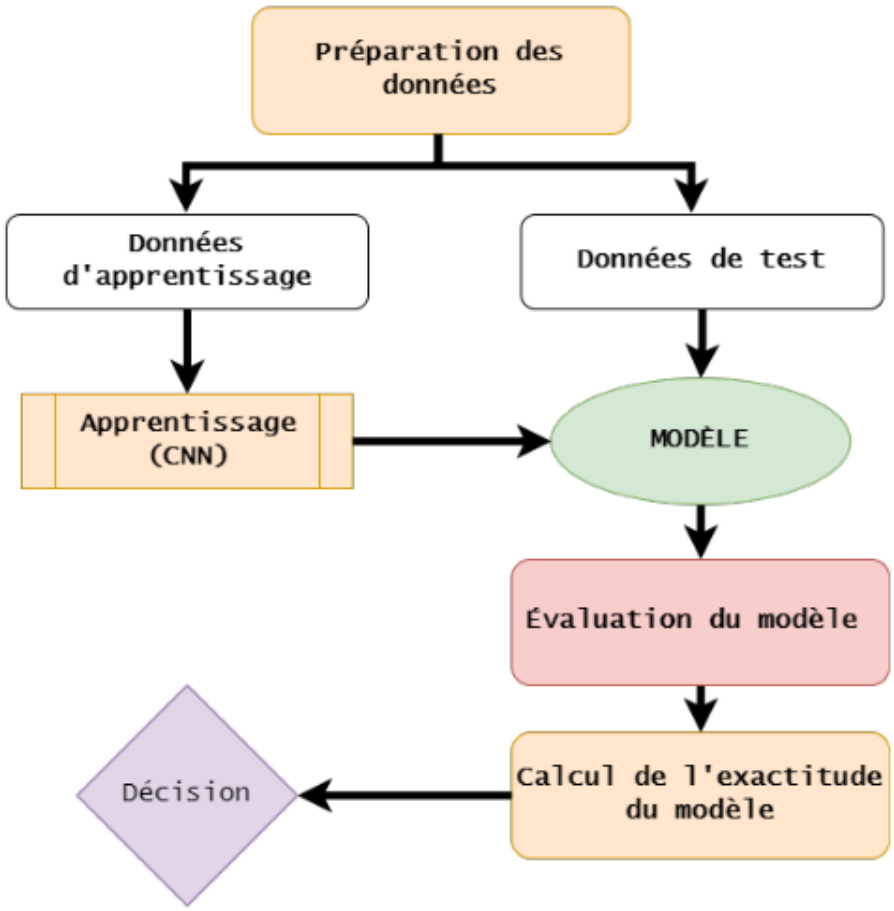


Figure III.1 — — Méthodologie de travail suivie pour exécuter un processus d'apprentissage.

III.2.1.1 Description du jeu de données GWCMC

Le jeu de données du Guangzhou Women and Children's Medical Center (GWCMC) contient 5863 images radiographiques en format JPEG. Ces images sont divisées en deux catégories : *NORMAL* et *PNEUMONIA*. Les images ont été sélectionnées à partir de radiographies thoraciques de patients pédiatriques âgés de un à cinq ans du GWCMC.

— **NORMAL** : Radiographies de patients sans signes de pneumonie.

— **PNEUMONIA** : Radiographies montrant des signes de pneumonie, qui peuvent être d'origine bactérienne ou virale.

Les images sont annotées par des radiologistes experts, garantissant la qualité et la fiabilité des données pour les applications de détection de la pneumonie. La Fig. III.2 et la Fig. III.2 illustre des exemples d'images issues de ce jeu de données avec leurs étiquettes.

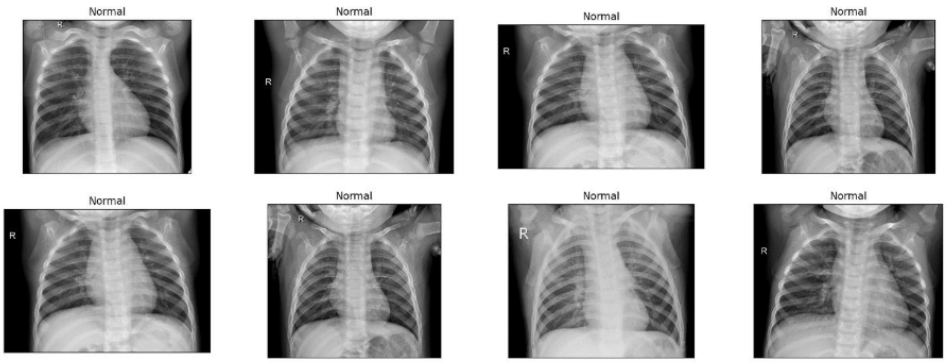


Figure III.2 — — Échantillons d'images tirés du jeu de données GWCMC de la classe NORMAL.

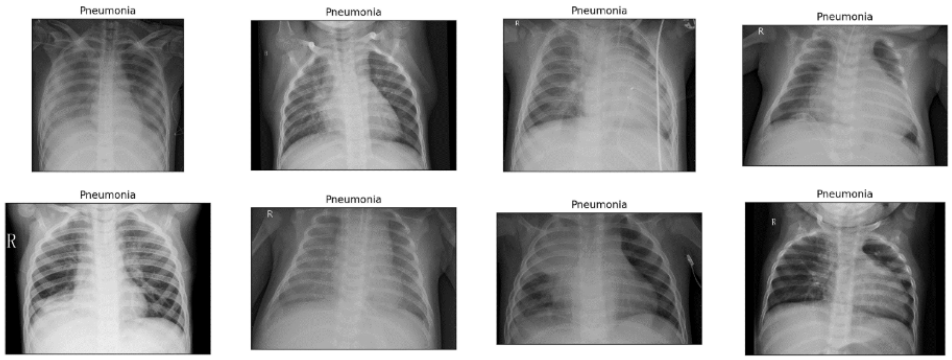


Figure III.3 — — Échantillons d'images tirés du jeu de données GWCMC de la classe PNEUMONIA.

III.2.1.2 Répartition des Données dans le Jeu de Données GWCMC

Le jeu de données GWCMC est divisé en deux ensembles principaux : les données d'entraînement et les données de test. Cette répartition est essentielle pour évaluer la performance des modèles de manière objective et garantir que les modèles ne sont pas surajustés aux données d'entraînement.

Ensemble d'Entraînement

L'ensemble d'entraînement est utilisé pour former les modèles de détection de la pneumonie. Il contient 5216 images, dont :

— 1341 images **NORMAL**

— 3875 images **PNEUMONIA**

L'ensemble d'entraînement est crucial pour permettre aux modèles d'apprendre les caractéristiques distinctives des radiographies normales et de celles présentant des signes de pneumonie.

Ensemble de Test

L'ensemble de test est utilisé pour évaluer la performance des modèles une fois qu'ils ont été formés. Il contient 624 images, dont :

— 234 images **NORMAL**

— 390 images **PNEUMONIA**

L'ensemble de test permet de mesurer la précision, le rappel, et le F-score des modèles, fournissant une évaluation objective de leur performance sur des données non vues auparavant.

Visualisation de la Répartition des Données

La Fig. III.4 ci-dessous illustre la répartition des données entre les ensembles d'entraînement et de test :

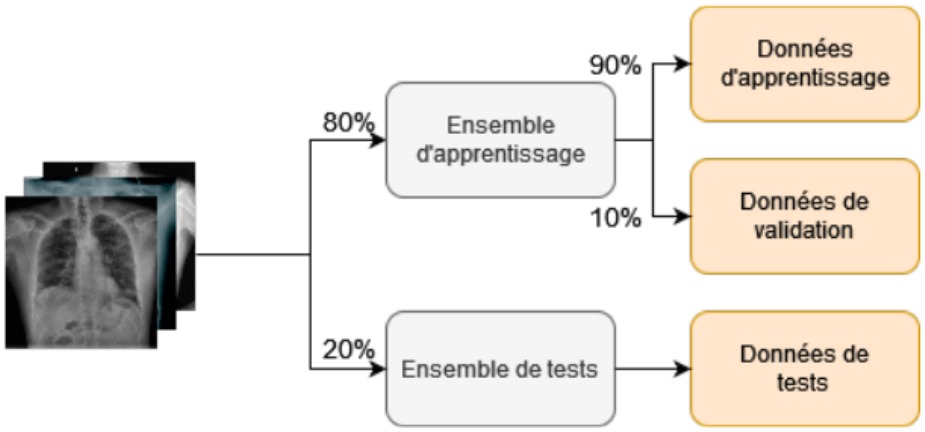


Figure III.4 — — Répartition des données en un ensemble d'apprentissage et de tests.

Cette répartition est conçue pour assurer que les modèles formés sont capables de généraliser à de nouvelles données, garantissant ainsi leur efficacité et leur robustesse dans des applications réelles. Les proportions équilibrées entre les classes **NORMAL** et **PNEUMONIA** dans les ensembles d'entraînement et de test sont essentielles pour éviter les biais de classification et assurer une évaluation équitable des performances des modèles.

III.2.1.3 Augmentation des Données

Pour améliorer la robustesse et la généralisation des modèles de détection de la pneumonie, diverses techniques d'augmentation des données ont été appliquées aux images radiographiques du jeu de données GWCMC. Ces techniques incluent :

— **Rotation** : Les images sont tournées aléatoirement à différents angles pour simuler différentes orientations possibles des radiographies.

— **Redimensionnement** : Les images sont redimensionnées de manière aléatoire, modifiant légèrement leur échelle tout en conservant les proportions originales.

— **Retournement Horizontal et Vertical** : Les images sont retournées horizontalement et/ou verticalement pour augmenter la diversité des perspectives.

— **Recadrage** : Des sections aléatoires des images sont recadrées et redimensionnées à la taille d'entrée souhaitée.

— **Changements de Luminosité et de Contraste** : Les niveaux de luminosité et de contraste des images sont ajustés de manière aléatoire pour simuler les variations de conditions de prise de vue.

Ces techniques d'augmentation des données sont appliquées de manière aléatoire pendant le processus d'entraînement, garantissant que chaque époque d'entraînement voit une variation différente des images, enrichissant ainsi le processus d'apprentissage du modèle.

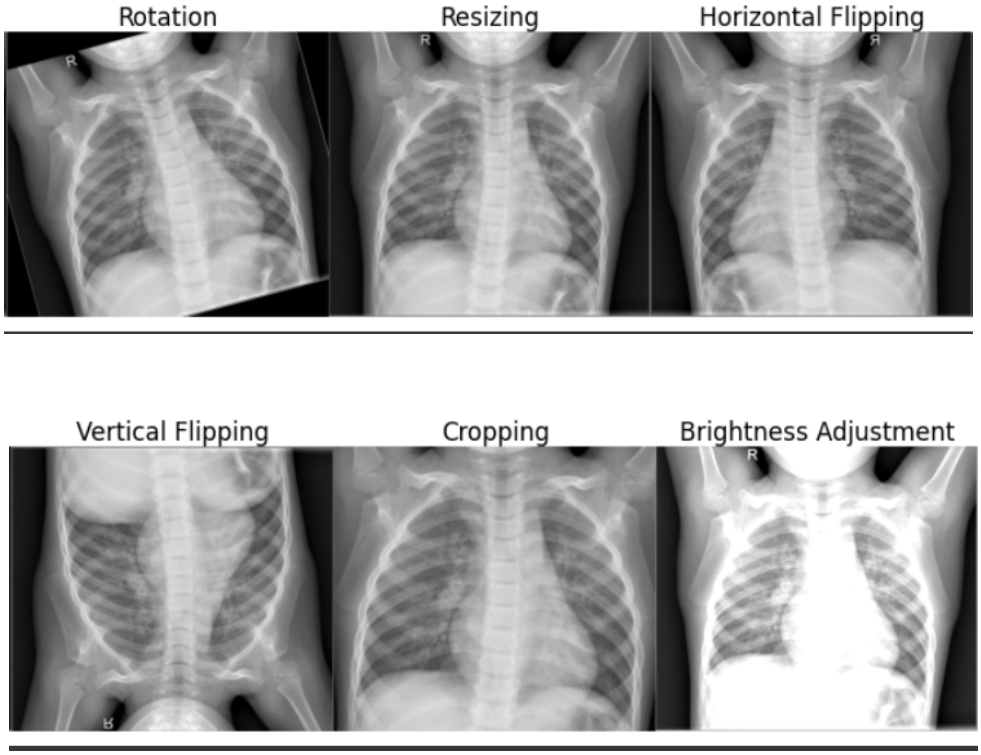


Figure III.5 — — Exemple des images générés par les techniques d'augmentations de données.

III.2.2 Architectures utilisées

Pour la détection de la pneumonie, nous avons utilisé plusieurs architectures de réseaux de neurones convolutifs (CNN) reconnues pour leurs performances élevées en classification d'images. Les architectures sélectionnées sont :

— **VGG16** : Créée par Simonyan et Zisserman en 2014 [96], VGG16 est une architecture CNN caractérisée par sa simplicité et sa consistance, utilisant des filtres de petite taille (3x3). Elle est connue pour ses bonnes performances en classification d'images.

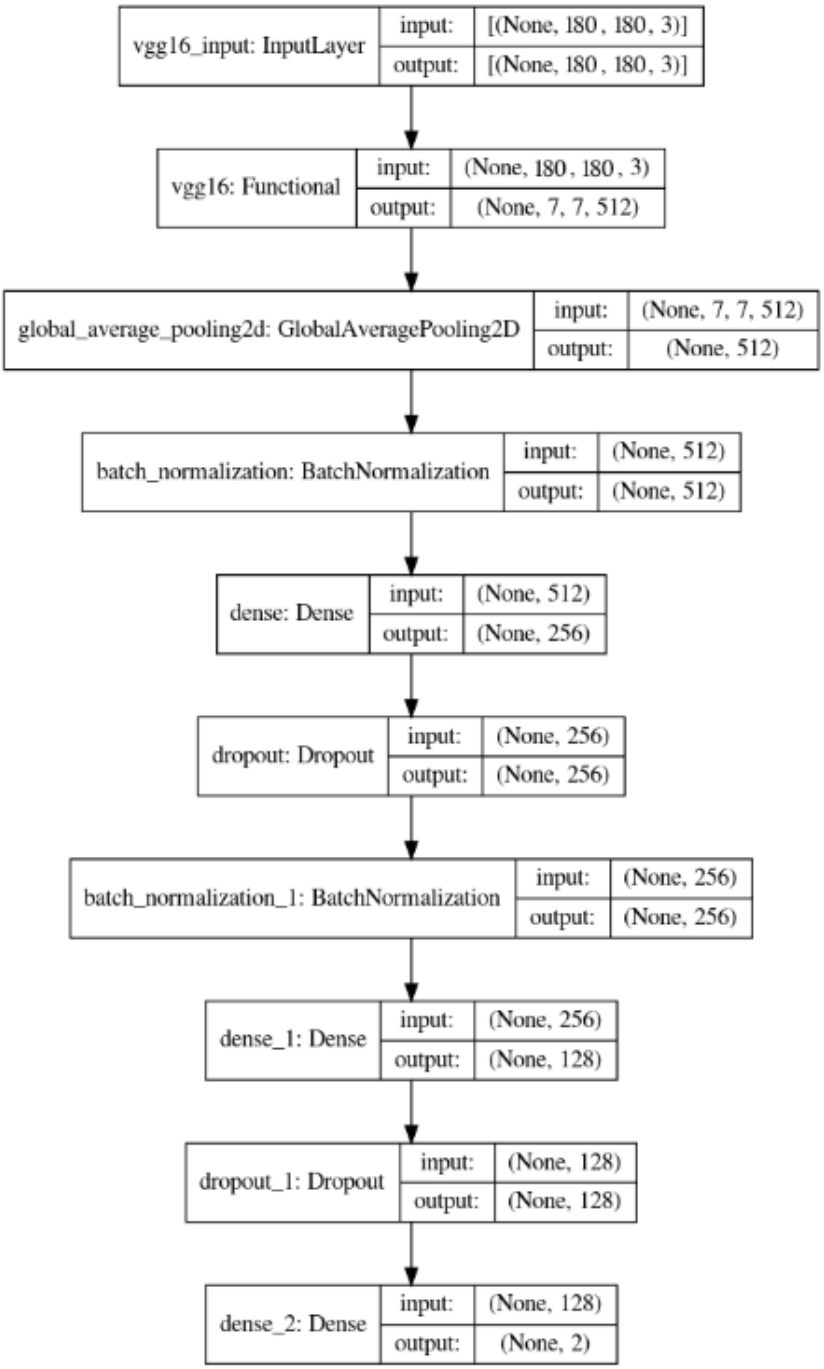


Figure III.6 — — Architecture VGG16.

— **ResNet50** : Introduit par He et al. en 2016 [97], ResNet50 est une architecture de réseau de neurones résiduel profond avec 50 couches. Elle utilise des connexions de saut pour permettre l'entraînement de modèles très profonds sans souffrir de la dégradation des performances.

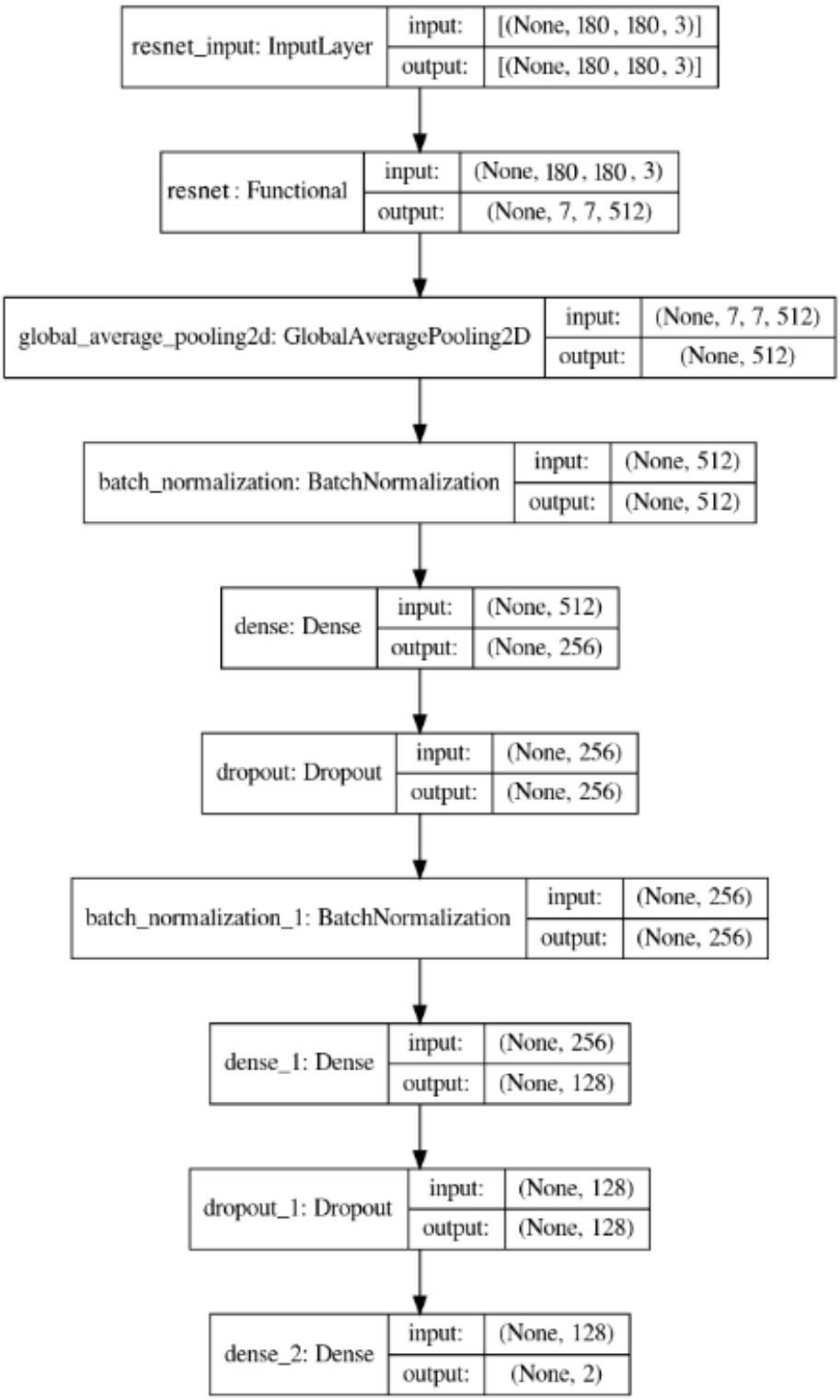


Figure III.7 — —Architecture ResNet50.

— **InceptionV3** : Développé par Szegedy et al. en 2015 [98], InceptionV3 équilibre la complexité du modèle avec l'efficacité computationnelle en utilisant des modules d'inception qui capturent des caractéristiques à différentes échelles.

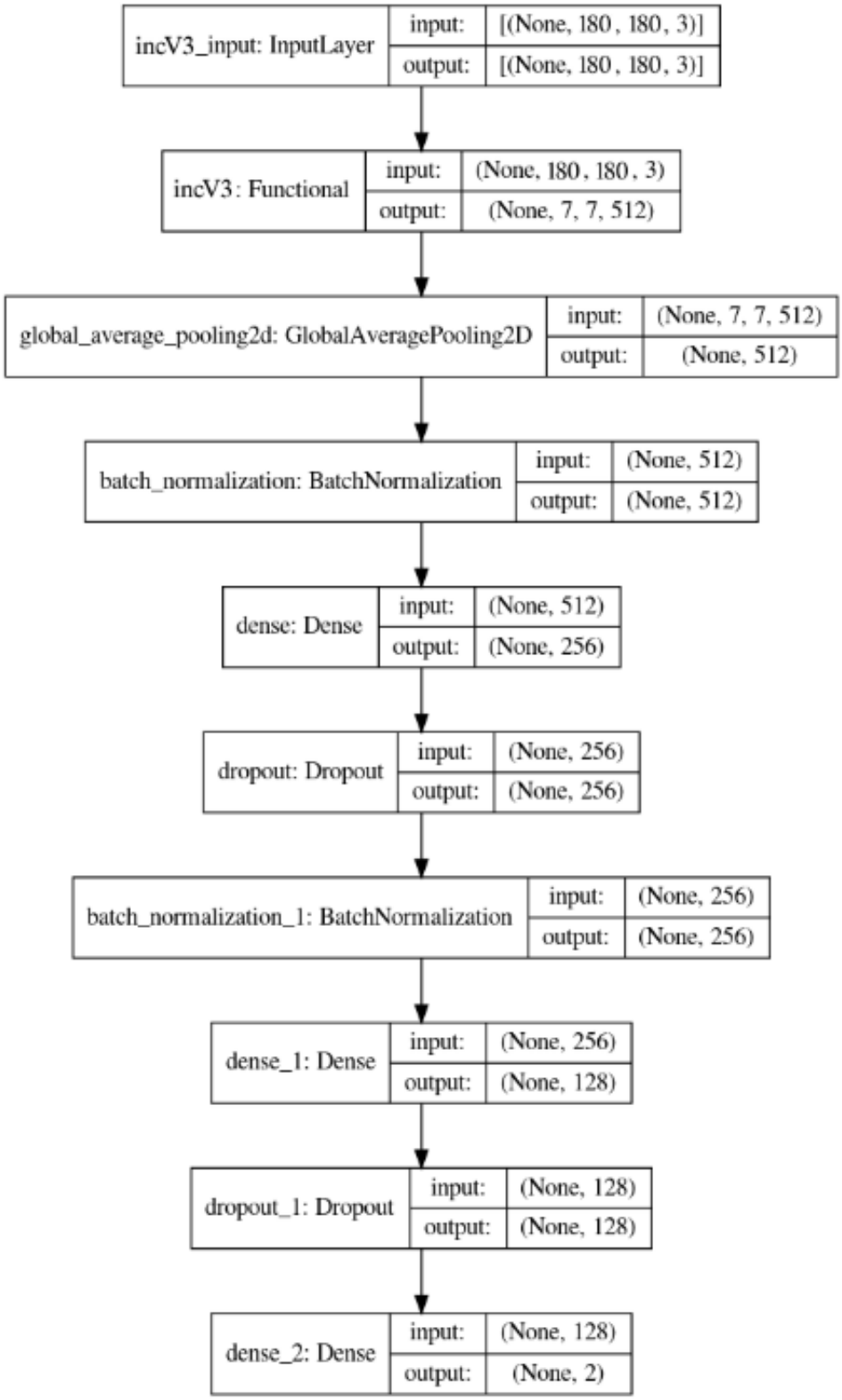


Figure III.8 — — Architecture InceptionV3.

III.2.3 Métriques d'évaluation

L'évaluation d'un modèle d'apprentissage consiste à tester son exactitude. Ceci revient à mesurer la distance entre les valeurs réelles dans les données de test et les valeurs prédites par ledit modèle. Dans notre cas où il s'agit d'un problème de classification, on définit

pour une classe A sa matrice de confusion, montrée dans la Fig. IV.8.

La matrice de confusion comporte les quatre notions suivantes :

— **Vrais positifs (True Positive (TP))** : sont les instances de A correctement prédites.

— **Faux positifs (False Positive (FP))** : sont les instances qui n'appartiennent pas à la classe A, mais elles sont prédites A.

— **Vrais négatifs (True Negative (TN))** : sont les instances non A et classées non A.

— **Faux négatifs (False Negative (FN))** : sont les instances de A, mais classées non A.

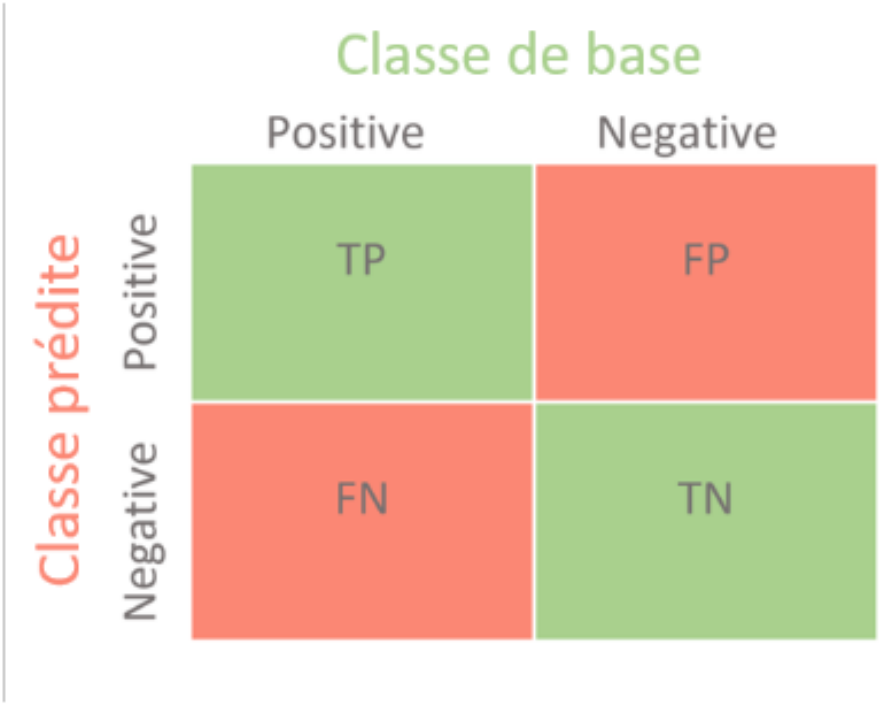


Figure III.9 — Matrice de confusion pour une classe A.

Pour évaluer la performance des modèles de détection de la pneumonie, plusieurs métriques d'évaluation ont été utilisées. Ces métriques permettent de quantifier la précision, la sensibilité et la spécificité des modèles. Les principales métriques d'évaluation utilisées sont :

III.2.3.1 Précision (Accuracy)

La précision est définie comme le rapport entre le nombre de prédictions correctes et le nombre total de prédictions. Elle mesure la proportion des prédictions correctes parmi toutes les prédictions effectuées par le modèle. La précision est calculée comme suit :

$$\text{Précision} = \frac{TP + TN}{TP + TN + FP + FN} \tag{III.1}$$

III.2.3.2 Précision

La précision est définie comme le rapport entre le nombre de vrais positifs et le nombre total de prédictions positives. Elle mesure la proportion des prédictions positives correctes parmi toutes les prédictions positives effectuées par le modèle. La précision est calculée comme suit :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{III.2}$$

III.2.3.3 Rappel (Sensibilité)

Le rappel, également connu sous le nom de sensibilité, est défini comme le rapport entre le nombre de vrais positifs et le nombre total de cas positifs réels. Il mesure la capacité du modèle à identifier correctement les cas positifs. Le rappel est calculé comme suit :

$$\text{Rappel} = \frac{TP}{TP + FN} \tag{III.3}$$

III.2.3.4 Score F1

Le score F1 est la moyenne harmonique de la précision et du rappel. Il fournit une mesure équilibrée de la performance du modèle, en tenant compte à la fois des faux positifs et des faux négatifs. Le score F1 est calculé comme suit :

$$\text{Score F1} = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{III.4}$$

III.2.3.5 AUC-ROC

La courbe ROC (Receiver Operating Characteristic) est une représentation graphique de la performance d'un modèle de classification binaire. L'aire sous la courbe ROC (AUC-ROC) quantifie la capacité du modèle à distinguer entre les classes positives et négatives. Un AUC-ROC proche de 1 indique une bonne performance du modèle. La courbe ROC est tracée en fonction du taux de vrais positifs (TPR) et du taux de faux positifs (FPR), définis comme suit :

$$\text{TPR} = \frac{TP}{TP + FN} \tag{III.5}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{III.6}$$

III.3 Configurations expérimentales

Dans cette section, nous décrivons les configurations suivies dans les tests expérimentaux. Nous allons détailler les expériences réalisées dans le mode d'apprentissage.

III.3.1 Application de l'apprentissage fédéré

Dans le but d'évaluer les techniques d'apprentissage fédéré dans le cadre de la classification de la pneumonie, nous avons effectué divers tests expérimentaux pour pallier les différents aspects discutés dans le Chapitre II. Pour cela, nous avons établi une architecture client/serveur que nous allons décrire par la suite ainsi que quatre configurations expérimentales différentes. Nous avons mis en œuvre un système d'apprentissage fédéré pour traiter les données médicales de manière décentralisée tout en respectant la confidentialité des patients. L'architecture client/serveur permet à chaque client (hôpital ou centre médical) de former un modèle local sur ses données, puis de partager uniquement les mises à jour du modèle (et non les données brutes) avec le serveur central. Ce dernier agrège les mises à jour reçues pour améliorer le modèle global.

III.3.1.1 Configuration du Réseau de la Plateforme

La configuration du réseau de la plateforme d'apprentissage fédéré repose sur une architecture client/serveur, conçue pour permettre une formation collaborative des modèles de classification de la pneumonie tout en garantissant la confidentialité des données des patients. Cette section détaille les composants et le flux de données au sein de cette architecture.

Architecture Client/Serveur

L'architecture client/serveur se compose des éléments suivants :

- **Clients** :
  - Les clients représentent les hôpitaux ou les centres médicaux disposant de données radiographiques locales. Chaque client dispose de son propre modèle de classification qu'il entraîne sur ses données locales.
  - Les clients effectuent les mises à jour du modèle localement, puis envoient uniquement les mises à jour des poids (et non les données brutes) au serveur central.
- **Serveur Central** :
  - Le serveur central reçoit les mises à jour des modèles de chaque client. Il agrège ces mises à jour pour améliorer le modèle global.

— Le serveur central redistribue le modèle global mis à jour à tous les clients pour une nouvelle itération d'entraînement local.

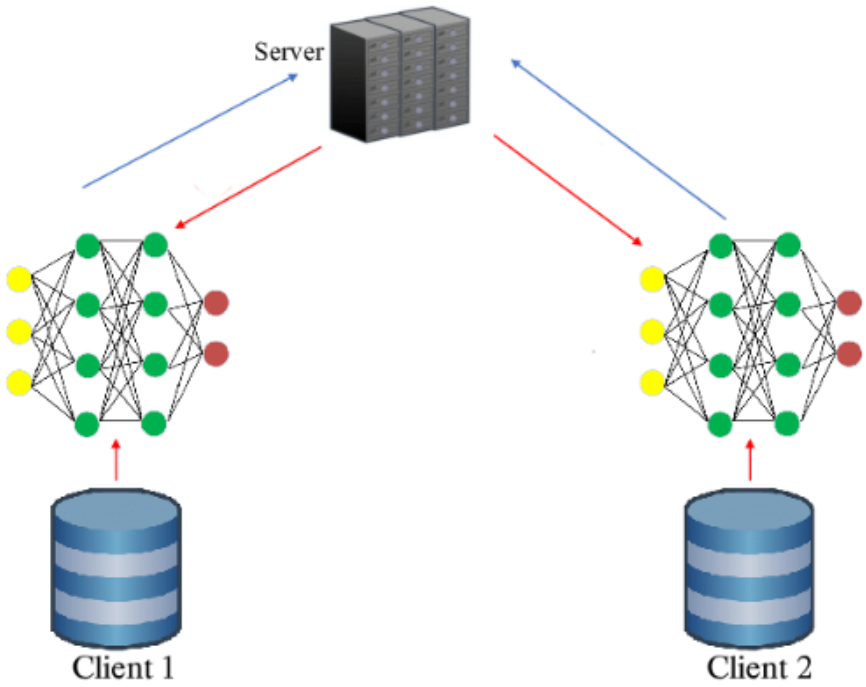


Figure III.10 — — Architecture du systeme de détéction de pneumonia.

Flux de Données

1. **Initialisation** :
   — Le serveur central initialise un modèle de base et le distribue à tous les clients participants.
2. **Entraînement Local** :
   — Chaque client entraîne le modèle initialisé sur ses données locales. Les données restent localement pour garantir la confidentialité.
3. **Mise à Jour des Poids** :
   — Après une période d'entraînement local, chaque client calcule les mises à jour des poids du modèle et les envoie au serveur central.
4. **Agrégation des Modèles** :
   — Le serveur central agrège les mises à jour des poids reçus de tous les clients pour mettre à jour le modèle global. Cette agrégation peut se faire via différentes méthodes, telles que l'agrégation fédérée moyenne (Federated Averaging).
5. **Distribution du Modèle Mis à Jour** :

— Le modèle global mis à jour est redistribué aux clients pour une nouvelle itération d'entraînement local.

Configurations Expérimentales

Trois configurations expérimentales ont été testées pour évaluer les performances de l'apprentissage fédéré :

1. **Synchronisation Complète** :
   — Tous les clients synchronisent leurs modèles avec le serveur central à chaque itération.
2. **Synchronisation Partielle** :
   — Seul un sous-ensemble des clients synchronise leurs modèles à chaque itération, tandis que d'autres clients effectuent plusieurs itérations locales avant de synchroniser.
3. **Entraînement Hiérarchique** :
   — Un modèle hiérarchique est utilisé où les clients sont regroupés en clusters. Chaque cluster a son propre modèle agrégé avant d'envoyer les mises à jour au serveur central.

Avantages de l'Architecture

— **Confidentialité des Données** :
   — Les données des patients ne quittent jamais les clients locaux, ce qui garantit la confidentialité et la sécurité des informations médicales.
— **Efficacité de l'Entraînement** :
   — L'entraînement fédéré permet de tirer parti des données disponibles dans plusieurs institutions sans nécessiter leur centralisation.
— **Scalabilité** :
   — L'architecture est scalable et peut être étendue pour inclure davantage de clients sans affecter de manière significative la performance du système.

la configuration du réseau de la plateforme d'apprentissage fédéré permet une formation collaborative efficace et sécurisée des modèles de classification de la pneumonie, tout en préservant la confidentialité des données des patients.

III.3.2 Description des configurations expérimentales

Les configurations expérimentales mises en place pour évaluer les performances de l'apprentissage fédéré dans la classification de la pneumonie sont conçues pour tester divers aspects de l'architecture client/serveur et des techniques de régularisation. Voici une description détaillée de ces configurations.

III.3.2.1 Configuration 1 : Synchronisation Complète

Dans cette configuration, tous les clients synchronisent leurs modèles avec le serveur central à chaque itération. Cette approche permet d'évaluer les performances de l'apprentissage fédéré lorsque toutes les mises à jour locales sont immédiatement agrégées dans le modèle global.

- **Processus** :
  - Chaque client entraîne le modèle localement sur ses propres données.
  - Les mises à jour des poids sont envoyées au serveur central après chaque itération.
  - Le serveur central agrège les mises à jour reçues et met à jour le modèle global.
  - Le modèle global mis à jour est redistribué à tous les clients pour la prochaine itération.
- **Avantages** :
  - Maximisation de la cohérence entre les modèles locaux et le modèle global.
- **Inconvénients** :
  - Communication fréquente et intensive entre les clients et le serveur central.

III.3.2.2 Configuration 2 : Synchronisation Partielle

Dans cette configuration, seul un sous-ensemble des clients synchronise leurs modèles à chaque itération, tandis que d'autres clients effectuent plusieurs itérations locales avant de synchroniser. Cela permet de réduire la charge de communication et d'évaluer l'impact de la synchronisation moins fréquente.

- **Processus** :
  - Un groupe de clients synchronise leurs modèles avec le serveur central après chaque itération.
  - Les autres clients synchronisent leurs modèles après plusieurs itérations locales.
- **Avantages** :
  - Réduction de la charge de communication.
  - Permet de tester la robustesse du modèle global avec des synchronisations moins fréquentes.
- **Inconvénients** :
  - Risque de divergence entre les modèles locaux et le modèle global.

III.3.2.3 Configuration 3 : Entraînement Hiérarchique

Dans cette configuration, les clients sont regroupés en clusters. Chaque cluster a son propre modèle agrégé avant d'envoyer les mises à jour au serveur central. Cela permet de tester une approche hiérarchique de l'apprentissage fédéré.

- **Processus** :
  - Les clients sont divisés en plusieurs clusters.
  - Chaque cluster entraîne et agrège un modèle local.
  - Les modèles agrégés des clusters sont ensuite envoyés au serveur central pour une agrégation globale.
- **Avantages** :
  - Réduction de la charge de communication entre les clients et le serveur central.
  - Permet une meilleure gestion de la scalabilité.
- **Inconvénients** :
  - Complexité accrue dans la gestion des clusters et de l'agrégation hiérarchique.

Ces configurations expérimentales permettent d'évaluer diverses approches et optimisations de l'apprentissage fédéré pour la classification de la pneumonie, en tenant compte des contraintes de communication, de scalabilité et de performance des modèles.

III.3.3 Bibliothèques et outils d'implémentation

Dans le cadre de l'implémentation de l'apprentissage fédéré pour la classification de la pneumonie, plusieurs bibliothèques et outils ont été utilisés pour faciliter le développement, l'entraînement et l'évaluation des modèles. Voici un aperçu des principaux outils et bibliothèques employés :

III.3.3.1 Bibliothèques de Deep Learning

- **TensorFlow** :
  - TensorFlow est une bibliothèque open-source de machine learning développée par Google. Elle offre une grande flexibilité pour la construction et l'entraînement de modèles de deep learning. TensorFlow est particulièrement adapté pour les tâches de classification d'images grâce à ses capacités avancées de manipulation de tenseurs et d'opérations mathématiques.
- **Keras** :

- Keras est une API de haut niveau pour le deep learning, construite sur TensorFlow. Elle permet de définir et d'entraîner des réseaux de neurones de manière intuitive et efficace. Keras simplifie la construction de modèles complexes et facilite leur entraînement et évaluation.

— **PyTorch** :

- PyTorch est une bibliothèque open-source de deep learning développée par Facebook. Elle est très populaire pour la recherche en machine learning grâce à sa flexibilité et à son mode d'exécution dynamique. PyTorch est utilisé pour construire et entraîner des modèles de réseaux de neurones, et offre des fonctionnalités avancées pour l'apprentissage fédéré.

III.3.3.2 Bibliothèques d'Apprentissage Fédéré

— **Federated Learning (TFF)** :

- TensorFlow Federated (TFF) est une bibliothèque open-source développée par Google pour l'implémentation de l'apprentissage fédéré. TFF fournit des abstractions pour définir des algorithmes d'apprentissage fédéré et pour simuler des environnements fédérés.

— **PySyft** :

- PySyft est une bibliothèque open-source pour l'apprentissage fédéré, le calcul multipartite (MPC) et l'apprentissage automatique sécurisé. Développée par OpenMined, PySyft étend PyTorch pour permettre la manipulation de tenseurs de manière sécurisée et privée.

— **Flower** :

- Flower est une bibliothèque open-source pour l'apprentissage fédéré qui supporte plusieurs backends de machine learning tels que TensorFlow, Keras et PyTorch. Flower est conçu pour être flexible, évolutif et simple à utiliser, facilitant l'implémentation de scénarios d'apprentissage fédéré.

III.3.3.3 Outils de Gestion et de Traitement des Données

— **Pandas** :

- Pandas est une bibliothèque open-source de manipulation et d'analyse de données pour le langage de programmation Python. Elle offre des structures de données et des outils d'analyse de données performants et faciles à utiliser.

— **NumPy** :

- NumPy est une bibliothèque fondamentale pour le calcul scientifique en Python. Elle prend en charge des tableaux et des matrices multidimensionnels ainsi que des fonctions mathématiques de haut niveau pour effectuer des opérations sur ces tableaux.

— **Matplotlib** :

- Matplotlib est une bibliothèque de traçage pour Python qui permet de créer des visualisations statiques, animées et interactives. Elle est utilisée pour générer des graphiques et des figures à partir des données analysées.

Ces bibliothèques et outils ont été essentiels pour l'implémentation, l'entraînement et l'évaluation des modèles d'apprentissage fédéré pour la classification de la pneumonie. Ils ont permis de créer un environnement de développement flexible et puissant, capable de traiter des données complexes et de générer des résultats robustes.

III.4 Discussion des Résultats

Ce chapitre présente une synthèse des résultats obtenus dans les sections précédentes de l'étude, en abordant divers aspects tels que la sélection des modèles, les métriques d'évaluation, la mise en œuvre de l'apprentissage fédéré, le déploiement web, et la conteneurisation avec Docker.

III.4.1 Performance des Modèles

Les performances des modèles VGG16, ResNet50 et InceptionV3 ont été évaluées selon plusieurs métriques. Malgré des valeurs de précision et de rappel élevées pour le VGG16, le modèle ResNet50 s'est révélé supérieur en termes de performance globale et de temps d'entraînement. ResNet50 a montré des valeurs compétitives de précision, de rappel et de F1 score, grâce à son utilisation efficace des couches convolutionnelles et des tailles de filtres adaptées. Ce modèle a su identifier avec précision les caractéristiques clés des radiographies thoraciques, essentielles pour détecter la pneumonie.

Modèle	Précision	Rappel	F1 Score
VGG16	0.89	0.90	0.91
ResNet50	0.87	0.92	0.89
InceptionV3	0.88	0.88	0.91

Table III.1 — Métriques des différents modèles

III.4.2 Apprentissage Fédéré

L'utilisation de l'apprentissage fédéré, avec la bibliothèque Flower, a permis un entraînement collaboratif et distribué. Chaque client a formé son modèle localement et a régulièrement envoyé les mises à jour au serveur central. Le serveur, utilisant l'algorithme Federated Averaging (FedAvg), a pu agréger les poids des modèles locaux pour créer un modèle global. Cette méthode a montré que l'apprentissage fédéré est une solution viable pour préserver la confidentialité des données tout en permettant un apprentissage efficace, particulièrement utile dans des contextes où la centralisation des données est problématique. La Figure III.11 illustre l'évolution de la précision du modèle global à travers les cycles d'entraînement.

III.4.3 Déploiement Web

Le système de détection de la pneumonie a été déployé via une application web utilisant Flask et Swagger. Cette combinaison a permis de créer une interface interactive et conviviale où les utilisateurs peuvent télécharger des images de radiographies thoraciques et obtenir des prédictions en temps réel. Flassger a été utilisé pour documenter et spécifier les points de terminaison de l'API, facilitant ainsi l'intégration et l'utilisation du service par les professionnels de santé et les chercheurs.

En résumé, les résultats de cette étude montrent que l'apprentissage fédéré, associé à des modèles de deep learning performants, offre une solution efficace et sécurisée pour la détection de la pneumonie à partir de radiographies thoraciques. Le déploiement web permet une utilisation pratique et accessible de ce système par les professionnels de la santé.

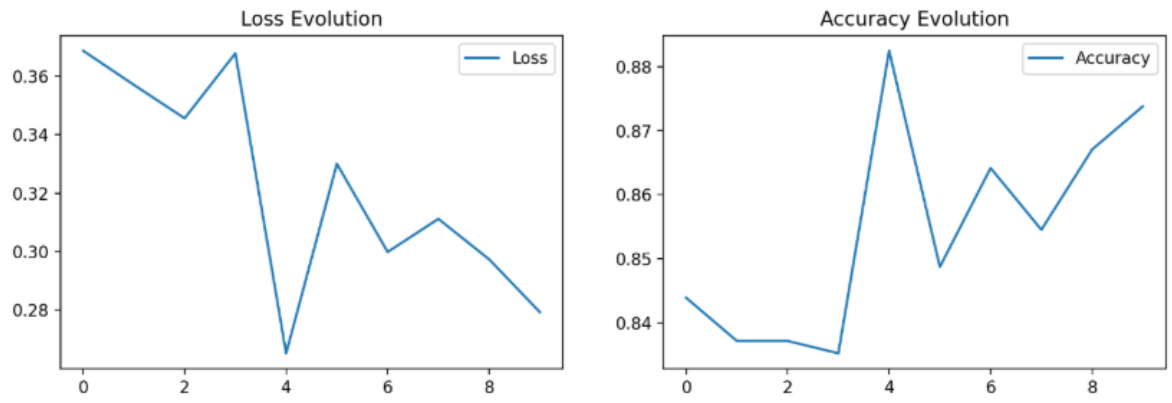


Figure III.11 — Évolution de la précision du modèle global

L'utilisation de Docker pour containeriser le système a amélioré sa portabilité et sa scalabilité. Le déploiement modulaire a été rendu possible grâce à des conteneurs Docker

indépendants pour le serveur, les clients et l'interface web. Le volume partagé pour le stockage des poids des modèles a maintenu la cohérence et l'accessibilité, et le fichier Docker Compose a facilité la gestion des conteneurs. L'architecture des conteneurs Docker du système est illustrée dans la Fig. III.12.

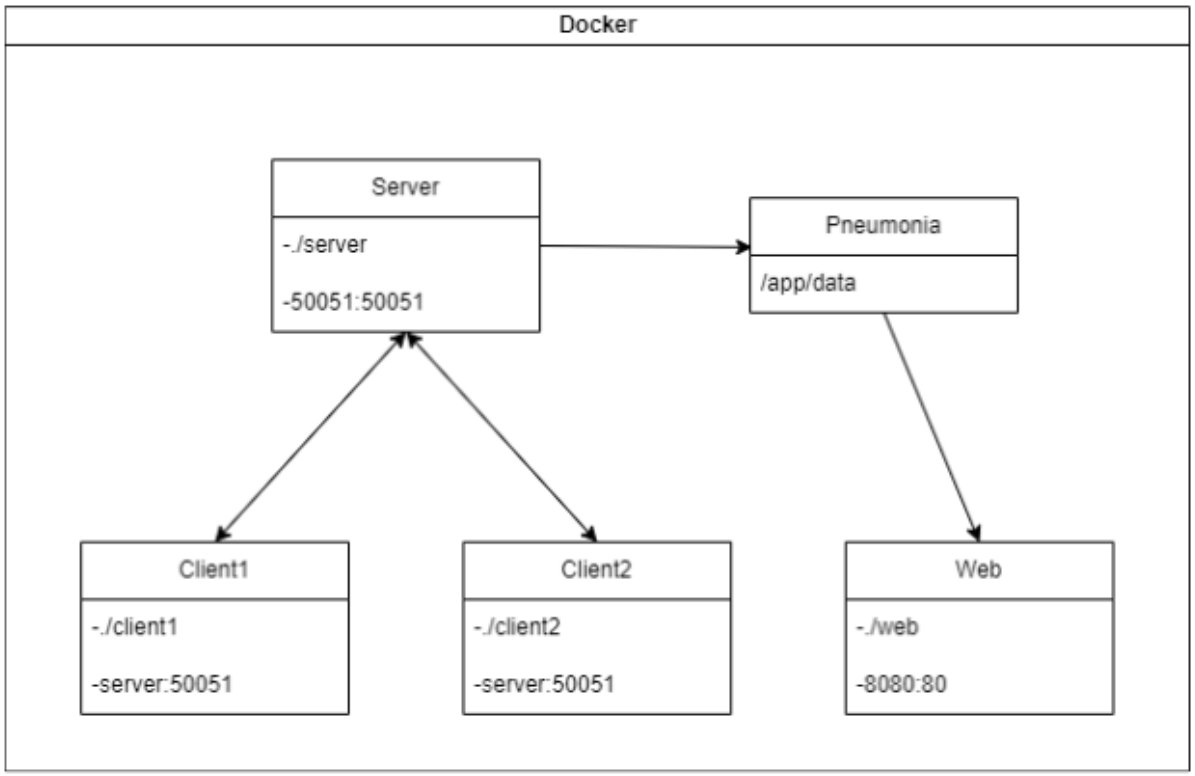


Figure III.12 — l'Architecture du docker

Pour mieux illustrer le déroulement du processus d'apprentissage fédéré, nous présentons le diagramme de séquence de la Fig. III.13. Ce diagramme décrit les interactions entre les différents composants du système, notamment les clients et le serveur central, tout au long des phases d'entraînement et d'agrégation. Chaque étape, depuis l'envoi des mises à jour du modèle local par les clients jusqu'à la consolidation des modèles sur le serveur central, est représentée de manière séquentielle. Ce diagramme permet de visualiser clairement le flux d'informations et les mécanismes de coordination nécessaires pour garantir l'efficacité et la confidentialité dans un environnement d'apprentissage fédéré.

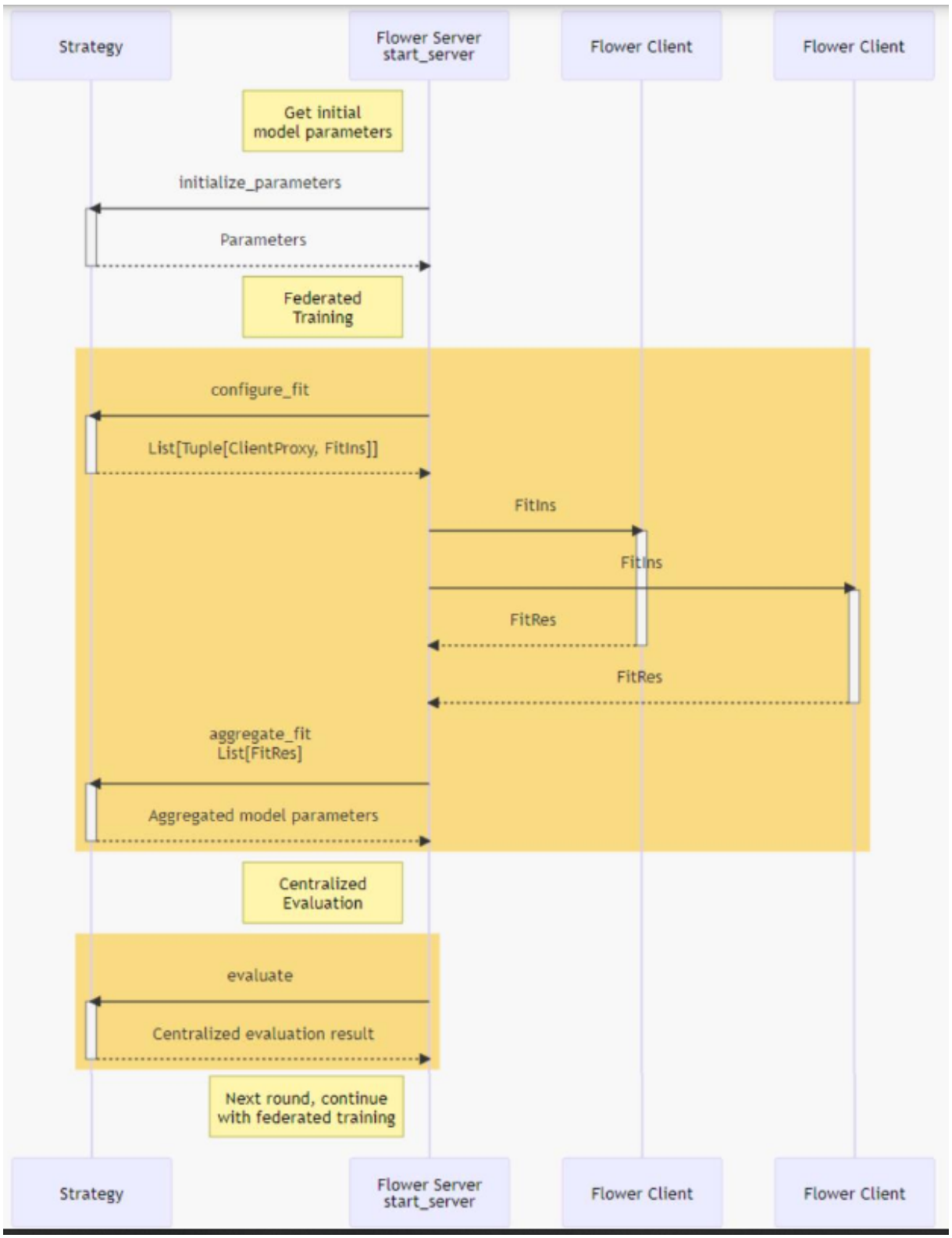


Figure III.13 — Diagramme de séquence du processus d'apprentissage fédéré

III.5 Performance Metrics des Modèles

Pour évaluer les performances des différents modèles, plusieurs métriques ont été utilisées, notamment l'exactitude, la précision, le rappel et le F1 score. Une matrice de

confusion a également été créée pour visualiser les résultats de classification. Ces données permettent de mieux comprendre la capacité des algorithmes à classer les patients atteints de pneumonie.

Une évaluation complète des performances du modèle a été réalisée sur l'ensemble des données, en se basant sur le modèle résultant de la dernière phase d'entraînement, qui intègre les contributions de tous les clients. Cette approche permet d'évaluer plus précisément la performance du modèle dans l'identification des cas de pneumonie en capturant les connaissances collectives des clients distribués.

Modèle	Précision	Rappel	F1-Score	Support
VGG16 (Normal)	0.89	0.83	0.86	235
VGG16 (Pneumonia)	0.90	0.94	0.92	391
Exactitude		0.90		
Macro Moy	0.90	0.89	0.89	626
Moy. Pondérée	0.90	0.90	0.90	626
ResNet50 (Normal)	0.80	0.88	0.84	235
ResNet50 (Pneumonia)	0.92	0.87	0.89	391
Exactitude		0.87		
Macro Moy	0.86	0.87	0.86	626
Moy. Pondérée	0.88	0.87	0.87	626
InceptionV3 (Normal)	0.89	0.80	0.84	235
InceptionV3 (Pneumonia)	0.89	0.94	0.91	391
Exactitude		0.89		
Macro Moy	0.89	0.87	0.88	626
Moy. Pondérée	0.89	0.89	0.89	626

Table III.2 — Métriques des différents modèles

Chaque modèle a été entraîné pendant 10 cycles d'apprentissage fédéré : VGG16 nécessitait environ 57 minutes par cycle, ResNet50 environ 31 minutes, et InceptionV3 environ 18 minutes. Après l'entraînement, nous avons utilisé le jeu de test pour évaluer les modèles et générer les résultats de classification.

— **VGG16** a montré des valeurs de rappel de 0.89 et 0.90 pour les classes normales et pneumonie respectivement, avec des valeurs de précision de 0.83 pour normal et 0.94 pour pneumonie, pour une exactitude globale de 0.90.

— **ResNet50** a obtenu des scores de précision de 0.80 pour normal et 0.92 pour pneumonie, avec des valeurs de rappel de 0.88 et 0.87 respectivement, pour une exactitude globale de 0.87.

— **InceptionV3** a produit des valeurs de précision de 0.89 pour les deux classes, avec des valeurs de rappel de 0.80 pour normal et 0.94 pour pneumonie, pour une exactitude globale de 0.89.

En conclusion, **ResNet50** a été choisi comme le modèle optimal pour l'identification de la pneumonie. Il a démontré un bon équilibre entre précision, rappel et exactitude, faisant de lui une option fiable pour le diagnostic de la pneumonie. Son architecture de réseau résiduel profond permet de capturer des détails fins et d'extraire des caractéristiques essentielles des radiographies thoraciques, ce qui est crucial pour une identification précise des patients atteints de pneumonie.

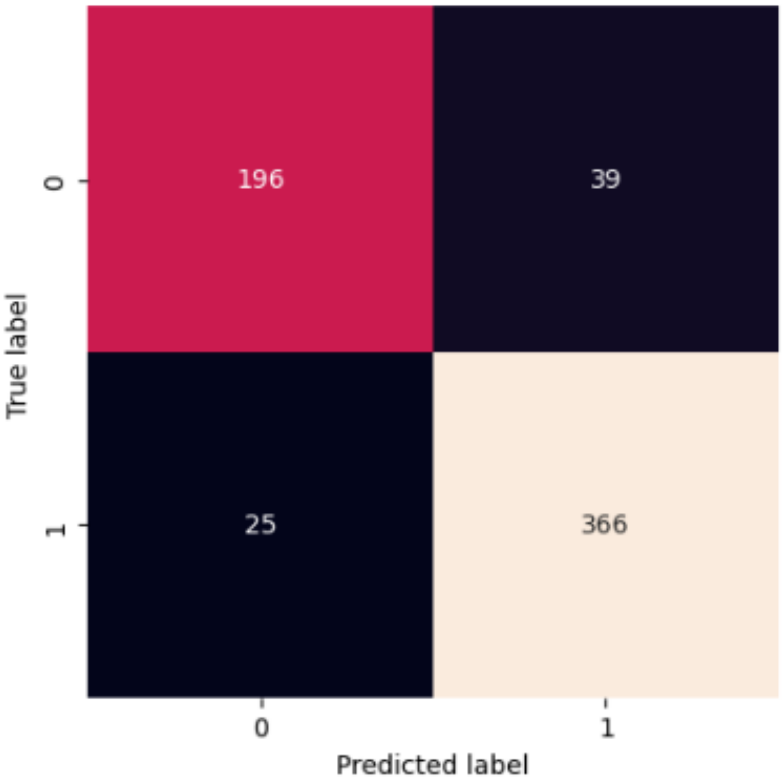


Figure III.14 — Matrice de confusion pour le modèle VGG16

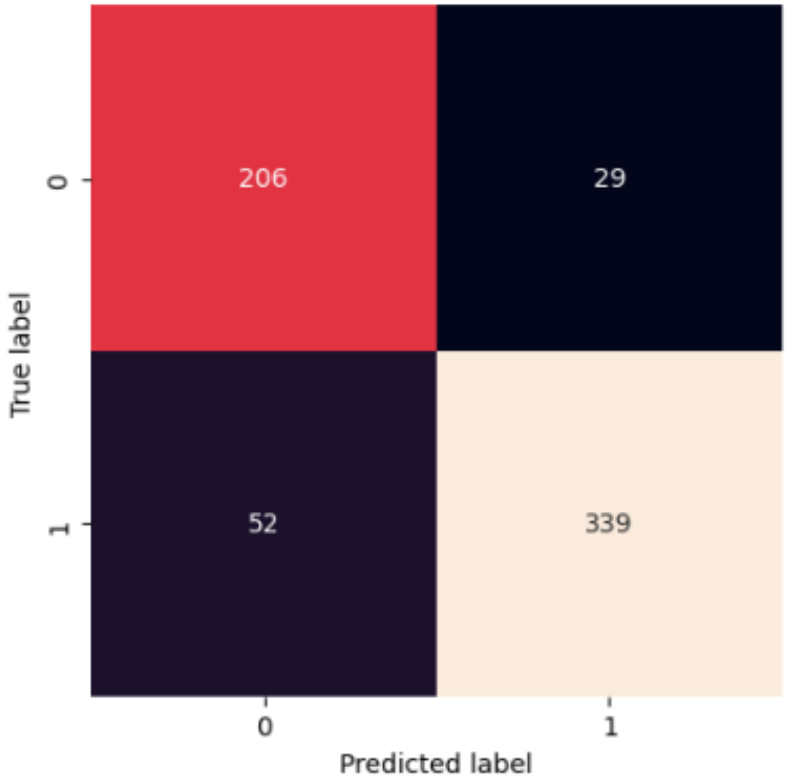


Figure III.15 — Matrice de confusion pour le modèle ResNet50

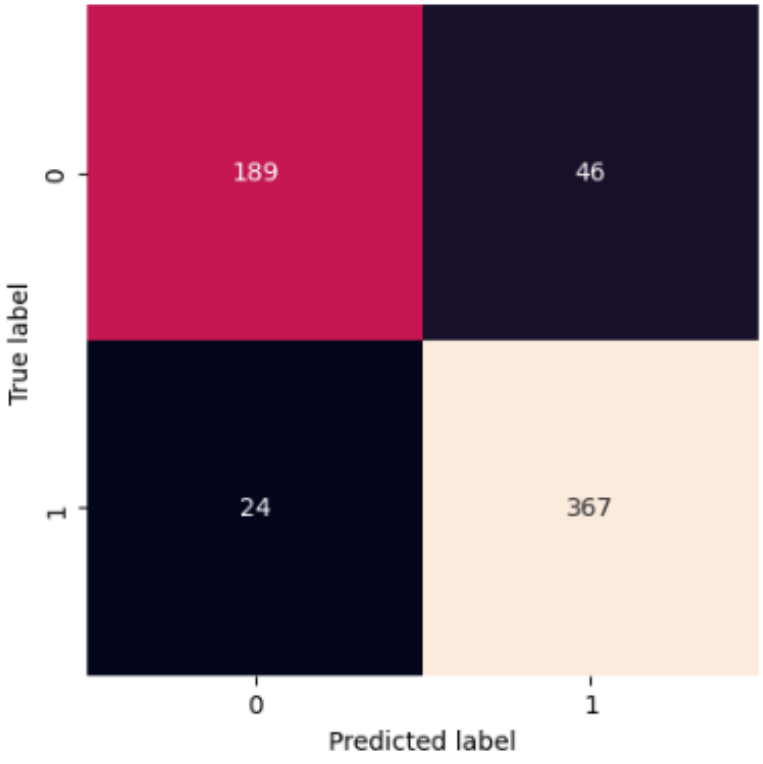


Figure III.16 — Matrice de confusion pour le modèle InceptionV3

III.6 Test de l'application web

Nous avons effectué des tests en téléchargeant 2 images de radiographies thoraciques (1 normales et 1 atteintes de pneumonie) et en comparant les prédictions avec les valeurs réelles pour évaluer la fonctionnalité de l'application web. Cette phase de test fournit des informations sur la fonctionnalité du système dans le monde réel et sur sa précision dans l'identification des cas de pneumonie. Nous examinerons les résultats prédits, évaluerons l'exactitude et discuterons des problèmes rencontrés pendant le processus de test.

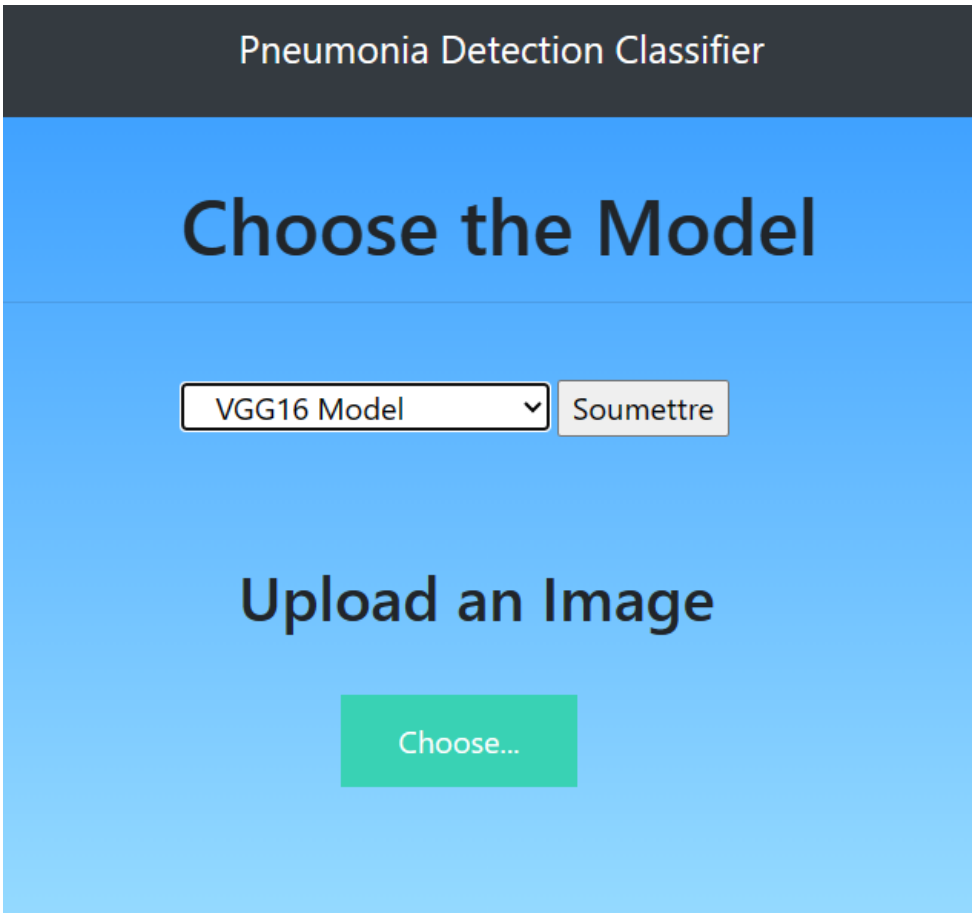


Figure III.17 — Présentation de l'interface de notre systeme

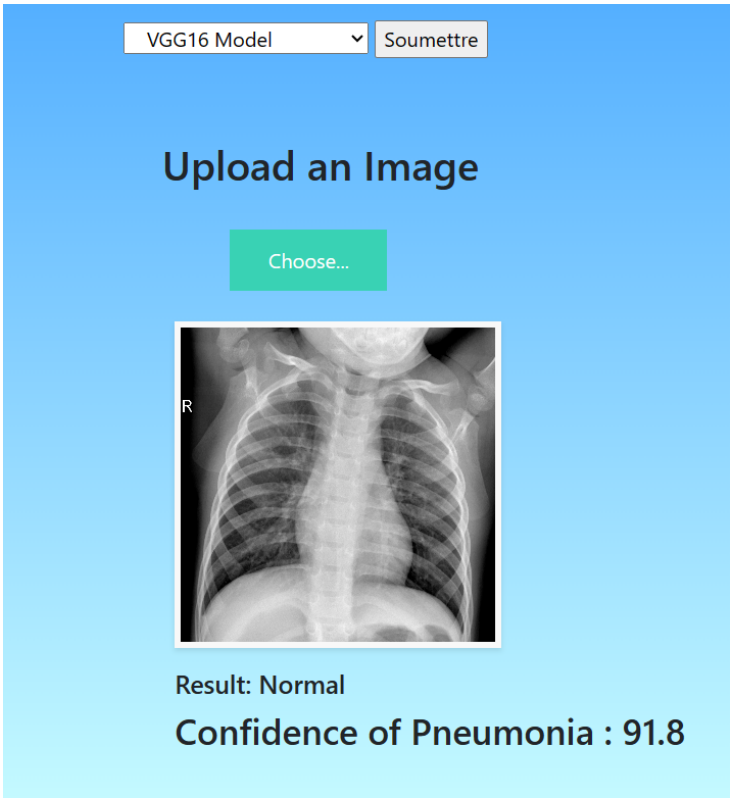


Figure III.18 — Exemple sur un patient dans un cas NORMAL

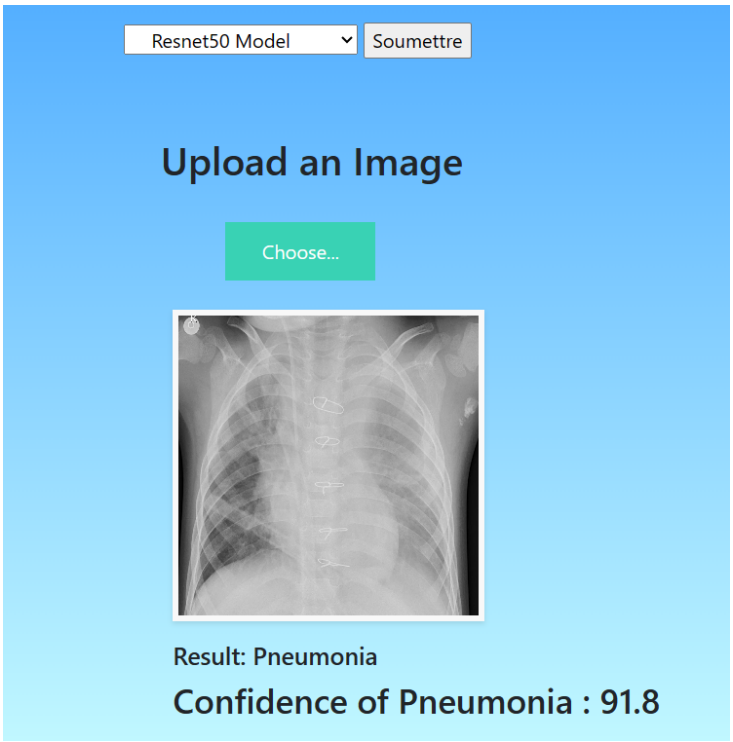


Figure III.19 — Exemple sur patient dans un cas PNEUMONIA

III.7 Conclusion

Ce chapitre a démontré que l'apprentissage fédéré est une méthode efficace pour entraîner des modèles de détection de la pneumonie tout en préservant la confidentialité des données des patients. Les performances des modèles VGG16, ResNet50 et InceptionV3 ont été évaluées, ResNet50 se distinguant comme le plus performant en termes d'équilibre entre précision, rappel et F1 score.

Les résultats montrent que l'apprentissage fédéré peut fournir des modèles fiables et précis pour le diagnostic médical, tout en facilitant la collaboration entre différentes institutions de santé sans compromettre la sécurité des données.

Conclusion générale et perspectives

L'intégration de l'intelligence artificielle (IA) dans le secteur médical représente une révolution qui transforme les pratiques diagnostiques et thérapeutiques. À travers l'exploration des multiples dimensions de l'IA, nous avons démontré comment les différentes formes d'apprentissage automatique, telles que l'apprentissage supervisé, non supervisé et par renforcement, contribuent à la création d'outils de diagnostic et de traitement modernes. Ces technologies ne se contentent pas d'améliorer les capacités médicales existantes mais personnalisent également les soins en offrant des traitements adaptés aux besoins individuels des patients.

Cependant, l'adoption de l'IA en médecine n'est pas exempte de défis, notamment les considérations éthiques et la confidentialité des données. L'apprentissage fédéré émerge comme une solution innovante pour surmonter ces obstacles en permettant la formation de modèles sur des données décentralisées, assurant ainsi la confidentialité et la sécurité des informations sensibles des patients.

En particulier, l'application de l'apprentissage fédéré dans la détection de pathologies, comme la pneumonie, a montré des résultats prometteurs. Les performances des modèles tels que VGG16, ResNet50 et InceptionV3 ont été évaluées, avec une mention spéciale pour ResNet50 qui a démontré un excellent équilibre entre précision, rappel et F1 score. Ces résultats soulignent le potentiel de l'apprentissage fédéré à fournir des modèles fiables et précis pour le diagnostic médical, tout en facilitant la collaboration entre différentes institutions de santé sans compromettre la sécurité des données.

En revanche, l'intelligence artificielle et l'apprentissage fédéré représentent des avancées transformatrices pour le domaine médical. En intégrant ces technologies, nous pouvons améliorer non seulement l'efficacité et la précision des diagnostics mais aussi assurer des soins de santé personnalisés et sécurisés. Ce projet ouvre la voie à de futures innovations et collaborations dans le secteur de la santé, promettant un avenir où les technologies intelligentes jouent un rôle crucial dans l'amélioration des résultats pour les patients.

Enfin, nous pensons que notre projet peut être amélioré de différentes manières. Du point de vue de systeme, nous envisageons de développer d'autres fonctionnalités comme :

— Mettre en place un système de gestion des dossiers électroniques des patients.

— Donner la possibilité au patient de consulter les résultats de prédiction par les modèles intelligents à travers une application mobile. Dans ce sens, il faut établir un système de médiation et d'intégration de données pour récupérer les informations issues des différents hôpitaux.

— Développer un module de fusion de données à base de la théorie de l'évidence pour la prise de décision concernant l'annotation des nouvelles images issues des dispositifs médicaux.

— Améliorer la rentabilité et la présentation de l'application web développée.

Dans le volet traitant de l'apprentissage fédéré, nous pensons que les améliorations suivantes sont intéressantes :

— Développer d'autres modèles d'apprentissage profond à base de diverses modalités de données médicales autres que les images radiographiques et étendre la tâche de détection sur un ensemble assez large de pathologies.

— Appliquer l'apprentissage fédéré vertical sur des données de natures hétérogènes.

— Appliquer des techniques d'optimisation de modèles pour réduire le coût de communication entre le serveur et les clients.

— Développer une technique de sélection de clients participant dans le processus d'entraînement à base de la qualité des données que possède chacun d'eux.

Bibliographie

[1] F. Ahamed, "A review on brain tumor segmentation based on deep learning methods with federated learning techniques," *elseveir*, 2023.

[2] A. Darche, "L'intelligence artificielle pour les nuls," *Gestion*, vol. 40, no. 2, pp. 112–114, 2015.

[3] T. Jo, *Machine Learning Foundations : Supervised, Unsupervised, and Advanced Learning.* Springer Nature, 2021.

[4] A. A. Patel, *Hands-on unsupervised learning using Python : how to build applied machine learning solutions from unlabeled data.* O'Reilly Media, 2019.

[5] A. Kumar, M. Farik, M. Amin, M. Sultan, and R. Mahmud, "Machine learning methods and applications : A review," *International Journal of Machine Learning and Computing*, vol. 12, no. 2, pp. 65–78, 2022.

[6] D. Graupe, *Principles of artificial neural networks.* World Scientific, 2013, vol. 7.

[7] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET).* IEEE, 2017, pp. 1–6.

[8] P. Kim, "Convolutional neural network," in *MATLAB deep learning.* Springer, 2017, pp. 121–147.

[9] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things : Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[10] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim, "Deep learning in medical imaging : general overview," *Korean journal of radiology*, vol. 18, no. 4, pp. 570–584, 2017.

[11] K. Uma, "Review of the application of ai in various medical domains," *International Journal of Advanced Research in Science, Communication and Technology*, 2023.

[12] S. Gao, "Discussion on the application of ai in diagnosis and treatment in clinical medicine," *Highlights in Science Engineering and Technology*, 2023.

[13] M. Rastogi, "Exploring the growth potential of ai in the health sector," *Indian Journal of Applied Research*, 2023.

[14] I. S. Galdames, "Examining the integration of ai in daily medical practices," *International Journal of Medical and Surgical Sciences*, 2023.

[15] V. Mehta, "How ai is revolutionizing health systems by enhancing patient-centric diagnoses and treatments," *Journal of Medical Research and Innovation*, 2023.

[16] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning : A comprehensive survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–36, 2023.

[17] S. Awan, F. Li, B. Lyu, X. Liu, P. P. Jayaraman, and T. Jung, "Distributed machine learning : A comprehensive survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 2, pp. 245–265, 2023.

[18] D. Ravi, N. Navon, and J. He, "Federated learning for healthcare : A survey," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–35, 2023.

[19] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization : Distributed machine learning for on-device intelligence," *arXiv preprint arXiv :1610.02527*, 2016.

[20] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing : A systematic survey," *Sensors*, vol. 22, no. 2, p. 450, 2022.

[21] Z. Qin, G. Y. Li, and H. Ye, "Federated learning and wireless communications," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134–140, 2021.

[22] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust federated learning with noisy communication," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3452–3464, 2020.

[23] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[24] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv :1811.03604*, 2018.

[25] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated meta-learning for fraudulent credit card detection," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4654–4660.

[26] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv :1901.09888*, 2019.

[27] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 11–20, 2020.

[28] A. Jiménez-Sánchez, M. Tardy, M. A. G. Ballester, D. Mateus, and G. Piella, "Memory-aware curriculum federated learning for breast cancer classification," *arXiv preprint arXiv :2107.02504*, 2021.

[29] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[30] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[31] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning : A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[32] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, and R. C. et al., "Advances and open problems in federated learning," *CoRR*, 2019.

[33] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy. found trends theor comput sci 9 (3/4) : 211–407," 2014.

[34] Y. Dong, X. Chen, L. Shen, and D. Wang, "Eastfly : Efficient and secure ternary federated learning," *Computers & Security*, vol. 94, p. 101824, 2020.

[35] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv :1711.10677*, 2017.

[36] P. Mohassel and Y. Zhang, "Secureml : A system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 19–38.

[37] V. Mugunthan, A. Polychroniadou, D. Byrd, and T. H. Balch, "Smpai : Secure multi-party computation for federated learning," in *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[38] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *2019 IEEE international conference on big data (Big Data)*. IEEE, 2019, pp. 2552–2559.

[39] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[40] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, "A syntactic approach for privacy-preserving federated learning," in *ECAI 2020*. IOS Press, 2020, pp. 1762–1769.

[41] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet : Secure and verifiable federated learning," *IEEE Trans. Inf. Forens. Secur.*, vol. 15, no. 1, pp. 911–926, 2020.

[42] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[43] P. Mothukuri, "Mothukuri v., parizi rm, pouriyeh s., huang y., dehghantanha a., srivastava g," *A survey on security and privacy of federated learning, Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[44] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[45] A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3329–3337.

[46] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[47] Z. Chai, H. Fayyaz, Z. Fayyaz, A. Anwar, Y. Zhou, N. Baracaldo, H. Ludwig, and Y. Cheng, "Towards taming the resource and data heterogeneity in federated learning." in *OpML*, 2019, pp. 19–21.

[48] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data : A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[49] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv :1806.00582*, 2018.

[50] X. Fang and M. Ye, "Robust federated learning with noisy and heterogeneous clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 072–10 081.

[51] J. Guo, Z. Liu, K.-Y. Lam, J. Zhao, Y. Chen, and C. Xing, "Secure weighted aggregation for federated learning," *arXiv preprint arXiv :2010.08730*, 2020.

[52] J. Zhao, X. Zhu, J. Wang, and J. Xiao, "Efficient client contribution evaluation for horizontal federated learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3060–3064.

[53] P. M. Mammen, "Federated learning : Opportunities and challenges," *arXiv preprint arXiv :2101.05428*, 2021.

[54] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 055–20 070, 2022.

[55] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv :1602.05629*, 2017.

[56] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning : Convergence analysis and power-of-choice selection strategies," 2020.

[57] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv :2010.13723*, 2020.

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

[59] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning : Submodularity and convergence analysis," in *ICML 2021 International Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2021.

[60] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems : vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[61] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning : Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[62] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[63] K. Wei, J. Li, C. Ma, M. Ding, S. Wei, F. Wu, G. Chen, and T. Ranbaduge, "Vertical federated learning : Challenges, methodologies and experiments," *arXiv preprint arXiv :2202.04309*, 2022.

[64] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[65] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 48–54, 2020.

[66] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *2013 IEEE symposium on security and privacy*. IEEE, 2013, pp. 334–348.

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[68] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[69] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, "Open-source federated learning frameworks for iot : A comparative review and analysis," *Sensors*, vol. 21, no. 1, p. 167, 2020.

[70] A. Durrant, M. Markovic, D. Matthews, D. May, J. Enright, and G. Leontidis, "The role of cross-silo federated learning in facilitating data sharing in the agri-food sector," *Computers and Electronics in Agriculture*, vol. 193, p. 106648, 2022.

[71] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning. arxiv 2019," *arXiv preprint arXiv :1912.04977*, 1912.

[72] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning : Improving google keyboard query suggestions," *arXiv preprint arXiv :1812.02903*, 2018.

[73] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale : System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[74] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv :1903.03934*, 2019.

[75] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 509–517.

[76] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," 2019.

[77] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020.

[78] S. M. Rostami, S. Samet, and Z. Kobti, "A study of blockchain-based federated learning," *Federated and Transfer Learning*, vol. 27, p. 139, 2022.

[79] R. Wang and W.-T. Tsai, "Asynchronous federated learning system based on permissioned blockchains," *Sensors*, vol. 22, no. 4, p. 1672, 2022.

[80] G. Zyskind, O. Nathan *et al.*, "Decentralizing privacy : Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*. IEEE, 2015, pp. 180–184.

[81] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "{Bitcoin-NG} : A scalable blockchain protocol," in *13th USENIX symposium on networked systems design and implementation (NSDI 16)*, 2016, pp. 45–59.

[82] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning : A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[83] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks : A contract theory approach," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 2019, pp. 1–5.

[84] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[85] Y. Tian, Y. Sun, and W. Yin, "Feddane : A federated newton-type method," *arXiv preprint arXiv :2001.01920*, 2020.

[86] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold : Stochastic controlled averaging for on-device federated learning." 2019.

[87] Y. Venkatesha, Y. Kim, H. Park, Y. Li, and P. Panda, "Addressing client drift in federated continual learning with adaptive optimization," *arXiv preprint arXiv :2203.13321*, 2022.

[88] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smithy, "Feddane : A federated newton-type method," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1227–1231.

[89] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g : Applications, challenges, and opportunities," *Engineering*, 2021.

[90] M. Alazab, S. P. RM, M. Parimala, P. Reddy, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity : concepts, challenges and future directions," *IEEE Transactions on Industrial Informatics*, 2021.

[91] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang, "Federated learning meets natural language processing : A survey," *arXiv preprint arXiv :2107.12603*, 2021.

[92] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated learning*. Springer, 2020, pp. 240–254.

[93] B. Yuan, S. Ge, and W. Xing, "A federated learning framework for healthcare iot devices," *arXiv preprint arXiv :2005.05083*, 2020.

[94] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth : A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[95] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[96] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv :1409.1556*, 2014.

[97] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[98] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv :1512.00567*, 2015. [Online]. Available : https://arxiv.org/abs/1512.00567

Résumé : L'intelligence artificielle (IA) révolutionne les soins de santé en offrant des solutions précises et efficaces. Cette recherche développe un systeme collaborative de détection des pathologie utilisant l'apprentissage fédéré, qui entraîne des modèles d'IA sur des données décentralisées, préservant ainsi la confidentialité et la sécurité des données. L'apprentissage fédéré améliore la détection précoce des ces pathologie, améliorant les résultats pour les patients en permettant au modèle d'apprendre à partir de plusieurs clients sans partager de données brutes. Cette étude aborde des défis tels que l'hétérogénéité des données et l'efficacité de la communication. Les résultats expérimentaux montrent une grande précision dans la détection des tumeurs, soulignant le potentiel de l'apprentissage fédéré dans le domaine médical.

Mots-clés : Intelligence Artificielle (IA), Apprentissage fédéré, Sécurité, Apprentissage Collaboratif, Confidentialité des données, Fiabilité, Confidentialité des Données.

Abstract : Artificial Intelligence (AI) is revolutionizing healthcare by offering precise and efficient solutions. This research develops a collaborative system for pathology detection using federated learning, which trains AI models on decentralized data, thus preserving data privacy and security. Federated learning enhances early detection of these pathologies, improving patient outcomes by allowing the model to learn from multiple clients without sharing raw data. This study addresses challenges such as data heterogeneity and communication efficiency. Experimental results show high accuracy in pathology detection, highlighting the potential of federated learning in the medical field.

Keywords : Artificial Intelligence (AI), Federated Learning, Security, Collaborative Learning, Data Privacy, Reliability, Data Confidentiality.

ملخص : مع تزايد توافر مصادر الحصول على الصور،تحدث الذكاء الاصطناعي (اي) ثورة في الرعاية الصحية من خلال تقديم حلول دقيقة وفعالة. هذا البحث يطور نظامًا تعاونيًا للكشف عن الأمراض باستخدام التعلم الفيدرالي، الذي يدرب نماذج الذكاء الاصطناعي على بيانات لامركزية، مما يحافظ على خصوصية وأمان البيانات. يحسن التعلم الفيدرالي الكشف المبكر عن هذه الأمراض، مما يحسن نتائج المرضى من خلال السماح للنموذج بالتعلم من عدة عملاء دون مشاركة البيانات الخام. تتناول هذه الدراسة تحديات مثل تباين البيانات وكفاءة التواصل. تظهر النتائج التجريبية دقة عالية في الكشف عن الأورام، مما يبرز إمكانيات التعلم الفيدرالي في المجال الطبي.

الكلمات المفتاحية : الذكاء الاصطناعي (اي)، التعلم الفيدرالي، الأمن، التعلم التعاوني، خصوصية البيانات، الموثوقية، سرية البيانات.