Participation au développement de frameworks d'apprentissage fédéré

par Stevenson Pather

sous la direction de M. Marc TOMMASI (tuteur entreprise)

et Mme Patricia PLENACOSTE SAWCZACK (tutrice universitaire)

Licence 3 Informatique parcours Info

Rapport de stage de fin d'études - 30 mai 2022 au 29 juillet 2022

Centre de recherche Inria Lille Nord - Europe
Park Plaza, Parc scientifique de la Haute-Borne, 40 Av. Halley, 59650 Villeneuve-d'Ascq

Université de Lille
42 Rue Paul Duez, 59000 Lille

Table des matières

1 Remerciements **1**

2 Résumé **2**

3 Abstract **3**

4 Introduction **4**

5 Contexte **5**
5.1 INRIA 5
5.2 Activité de l'INRIA 5
5.3 Equipe Magnet 6
5.4 Activité de Magnet 6
5.5 Stage proposé 7
5.6 Missions accomplies 7

6 Contribution **8**
6.1 Introduction du projet : Declearn 8
6.1.1 Fonctionnement du modèle d'apprentissage fédéré 8
6.1.2 Communication et contraintes 10
6.1.3 Structure de Declearn 10
6.1.4 Chaîne de production du modèle 11
6.1.4.1 Planificateur de tâches et scripts bash 11
6.1.4.2 Inconvénients 13
6.2 Amélioration des entrées-sorties du framework 13
6.2.1 Gestion des expériences 13
6.2.2 Organisation et envoi de la journalisation 14
6.2.3 Nouvelle structure de la chaîne de production 15

7 Développement personnel et professionnel **18**
7.1 Enjeux environnementaux du numérique 18

8 Conclusion **20**

9 Bilan **21**

10 Bibliographie **22**

11 Table des figures **23**

12 Table des algorithmes **24**

Remerciements

mes vifs remerciements s'adressent en tout premier lieu à mon maître de stage M. Marc Tommasi, directeur de l'équipe de recherche Magnet au sein de l'inria, pour avoir rendu ce stage possible auprès de son équipe durant neuf semaines et m'avoir permis de travailler sur un projet d'apprentissage machine. Je le remercie également pour le temps investi à mon encadrement ainsi que pour les échanges que nous avons pu avoir sur l'apprentissage fédéré et le métier de chercheur, ceux-ci ont été inspirant tant sur le plan professionnel que personnel.

Je remercie également Mme Nathalie VAUQUIER qui a participée à mon encadrement en se montrant disponible durant les réunions et pour répondre à mes questions sur le projet.

Je remercie également M. Paul ANDREY qui a rejoint l'équipe début juillet avec qui j'ai pu avoir des échanges enrichissants sur le projet et sur mon travail lors des réunions.

Plus largement, je remercie tous les membres de l'équipe Magnet avec qui j'ai pu échanger durant mon stage ainsi que le personnel de l'inria qui s'est montré accueillant à mon égard. L'environnement de travail et la qualité des échanges qui m'ont été offerts m'ont permis de mener à bien ce stage.

Enfin, je remercie ma tutrice universitaire Mme Patricia PLENACOSTE SAWCZACK pour son accompagnement et pour s'être montrée disponible dans les différentes démarches qui ont permis ce stage.

Résumé

Ce présent rapport constitue le mémoire de mon stage de neuf semaines au sein de l'institut national de recherche en informatique et en automatique dans l'équipe de recherche Magnet (MAchine learninG in information NETworks). Ce stage était dans le cadre de validation de la troisième année de Licence Informatique en parcours Info.

Ma mission durant ce stage fut de participer au développement d'un framework d'apprentissage fédéré implémenté en python par l'équipe Magnet qui sera rapidement mise en test au sein du groupement G4 d'hôpitaux de la région. Ce modèle d'apprentissage utilise les données de santé des patients de chaque hôpital afin de prédire, avec une certaine perte en fonction des paramètres, si un patient sera sujet à de nouvelles complications de santé en fonction de ses données médicales.

La première version de production du déploiement du modèle utilise des scripts bash pour exécuter les expériences avec le software et une crontab pour planifier les exécutions des scripts bash. Ces scripts bash posent plusieurs problèmes à l'équipe comme le manque de clarté de ceux-ci, les expériences précédemment exécutées le sont de nouveau dans les sessions suivantes enfin la transmission des résultats et de la journalisation qui est faite sur un dépôt git.

Ce document présente les solutions mises en oeuvres pour répondre à ces problèmes ainsi qu'aux attentes de l'équipe et des hôpitaux. En effet, une refonte des scripts est proposée en ajoutant une table d'expériences à exécuter et un fichier propre à chaque client qui contient l'identifiant de la dernière expérience exécutée sans erreurs, ce qui permet de ne plus avoir de clients qui relancent des expériences précédemment exécutées. Enfin l'ajout de redirection des traces d'exécutions python du software et des scripts bash vers des fichiers de logs, avec une arborescence horodatée locale pour ordonner par session et expériences les résultats et logs pour le transmettre au serveur maître en utilisant soit la communication établie pour l'expérience actuelle. Sinon en créant une nouvelle communication vers le serveur avec le protocole utilisé par le software pour forcer l'envoi en fin d'exécution des scripts.

Ces solutions permettent une simplification d'utilisation et de compréhension des scripts en plus de fournir une meilleure gestion de la journalisation en respectant les contraintes des hôpitaux.

Abstract

This report is the memoir of my nine-week internship at the National Institute for Research in Computer Science and Automation in the Magnet research team (MAchine learninG in information NETworks). This internship was part of the validation of the third year of undergraduate diploma in Computer Science.

My mission during this internship was to take part in the development of the federated learning framework implemented in python by the Magnet research team which will be quickly tested within the G4 group of hospitals in the region. This learning model uses patient health data from each hospital to predict, with some loss depending on the parameters, whether a patient will be prone to new health complications based on their medical data.

The first production version of the model deployment uses bash scripts to run experiments with the software and a crontab which is a task scheduler to plan their executions. These bash scripts are problematic for the team due to the lack of clarity, previously performed experiments are run again in subsequent sessions and the transmission of results and logging is done using a git repository.

This document shows the solutions implemented to solve these issues and meet the expectations of the team and hospitals. Indeed, a redesign of scripts is proposed by adding an experiment table to execute and untracked file for each hospital which contains the id of the last experiment executed without errors in order to avoid hospitals rerun experiments already run in previous sessions. Finally, the addition of redirection of scripts and software to log files, with a local time-stamped tree so as to order these by session and experiments will constitute the new organization of logging.

These solutions allow a simplification and more understandable of scripts, in addition to providing a better logging management while respecting the constraints of hospitals.

Introduction

Dans le cadre de mes études en Licence 3 Informatique parcours Info, à l'université de Lille, j'ai effectué un stage de neuf semaines au sein de l'équipe de recherche Magnet à l'inria [1], ce qui m'a permis de découvrir le travail dans un laboratoire de recherche en informatique, d'apprendre le fonctionnement d'un projet d'apprentissage machine, d'acquérir de nouvelles compétences techniques ou encore de mieux échanger en équipe sur mon travail pour le soumettre.

La recherche en informatique est un domaine qui me captive depuis quelques années que ce soit au travers de vulgarisation, articles et conférences. C'est pourquoi ce stage au sein d'une équipe de recherche fut pour moi une occassion rêvée de découvrir ce secteur d'activité. D'autant plus que le champ d'études de l'équipe Magnet est l'apprentissage machine qui à la suite de ce stage sera ma spécialité de master. Ce stage est donc en cohérence avec mon projet d'étude et professionnel.

Le modèle décentralisé développé par l'équipe en collaboration avec quatres hôpitaux de la région impose un certains nombre de contraintes pour permettre de garantir la sécurité et confidentialité des données personnelles des patients des hôpitaux, qui sont des données sensibles soumises à des lois qui encadrent leur manipulation. Comme notamment les protocoles de communication autorisés ainsi que la communication entre les hôpitaux et le serveur qui ne peut être établir qu'à l'iniation des machines des hôpitaux. C'est dans ce cadre qu'il m'a été demandé d'améliorer les entrées-sorties pour faciliter le déploiement et la récolte des résultats.

L'objectif était tout d'abord de comprendre le projet existant et le fonctionnement de celui-ci pour ensuite proposer une amélioration qui répondent aux attentes de l'équipe et des hôpitaux. Cela passera par une refonte des scripts en ajoutant une table d'expériences ainsi qu'une journalisation par session et expériences pour un meilleur suivi des ceux-ci.

Le présent rapport trace les phases du développement des améliorations. Il sera consistué de plusieurs chapitres. En premier lieu, une présensation du contexte qui décrira l'inria l'organisme d'accueil ainsi que l'équipe Magnet, puis une description du stage proposé et les missions accomplies durant le stage. Un second chapitre sera consacré aux contributions à savoir les missions réalisées. Un troisième chapitre portera sur les enjeux environnementaux du numérique dans le cadre de l'UE DPP. Puis un quatrième chapitre présentera la conclusion de ce rapport. Finalement, un dernier chapitre sera dédié au bilan de ce stage.

1. Institut national de recherche en informatique et en automatique

Contexte

5.1 INRIA

L'Institut national de recherche en informatique et en automatique fut créé le 3 janvier 1967 sous le nom de IRIA (Institut de recherche en informatique et en automatique) dans le cadre du plan Calcul et a pour mission le développement de la recherche et de la valorisation en sciences et techniques de l'information et de la communication, au niveau national comme au plan international. L'institut mène la stratégie nationale française de recherche en intelligence artificielle. Il est actuellement sous la direction de M. Bruno Sportisse depuis 2018. L'inria en quelques chiffres :

— 3900 scientifiques
— 200 équipes-projets dont 80% communes avec leurs partenaires
— 200 startups depuis 1984
— 10 centres de recherche au coeur des grandes universités de recherche

Le centre Inria de l'Université de Lille, créé en 2008, est implanté sur deux sites : à la Haute-Borne, au cœur du campus de l'université de Lille, et à EuraTechnologies, au sein de l'écosystème entrepreneurial. Le centre compte 15 équipes-projets. Son action mobilise plus de 360 personnes, scientifiques et personnels d'appui à la recherche et à l'innovation, issues de 38 nationalités. Le centre Inria de l'Université de Lille est sous la direction de Mme Mireille Régnier depuis octobre 2019. En quelques chiffres clés :

— 15 équipes-projets
— 10 startups issues des équipes-projets
— 1 politique de site ambitieuse

5.2 Activité de l'INRIA

Les axes scientifiques prioritaires du centre de recherche de Lille sont :

— Science des données
— Génie logiciel
— Systèmes cyberphysiques

Ces domaines sont ainsi les thématiques que l'on retrouve dans les 15 équipes-projets. Les travaux des équipes représente la majeure partie des activités de l'inria mais pas seulement. L'institut se veut être une plateforme ouvert aux échanges et aux partenariats, au cœur des écosystèmes académiques,

industriels et entrepreneuriaux du numérique, à l'avant-garde de la recherche et de l'innovation dans et par le numérique, avec une seule préoccupation : renforcer leur impact pour construire la souveraineté numérique par la recherche et l'innovation. L'institut possède notamment plusieurs axes de développement stratégique :

— Pérennisation de l'I-Site Université Lille Nord Europe (ULNE) et participation active à la construction de l'établissement cible, intégrant l'université de Lille, les écoles d'ingénieurs, le CHU et l'institut Pasteur de Lille
— Partenariat avec Euratechnologies pour la montée en gamme des technologies portées par les startups de l'incubateur
— Opération du projet régional humAIn en faveur de l'intelligence artificielle, en étroite articulation avec la région des Hauts-de-France
— Sciences du numérique appliquée aux projets de médecine personnalisée.

5.3 Equipe Magnet

L'équipe Magnet [1] est une équipe de recherche au centre INRIA Nord Europe de Lille et du laboratoire CRIStAL de l'université de Lille. Elle est actuellement composée de six chercheurs permanents :

— Aurélien Bellet
— Pascal Denis (Directeur Adjoint)
— Mikaela Keller
— Michaël Perrot
— Jan Ramon
— Marc Tommasi (Directeur)

Mais également sept ingénieurs, trois postdoctorants, neuf étudiants PhD, deux membres associés et une assistante administrative.

Le principal objectif de l'équipe Magnet est de rendre l'intelligence artificielle plus acceptable pour notre société notamment en répondant à des problèmes éthiques que peut poser l'apprentissage machine mais aussi en responsabilisant les utilisateurs finaux d'intelligences artificielles, leurs recherches sont centrées sur les questions de confidentialité, d'équité et sobriété des données.

5.4 Activité de Magnet

Leurs activités sont notamment l'étude des méthodes d'apprentissage automatique basées sur les graphes qui sont les fondements communs du groupe de recherche ainsi que des méthodes issues de la théorie de l'apprentissage statistique et computationnelle, de la théorie des graphes, de l'apprentissage des représentations, de l'optimisation (distribuée) et des statistiques. L'équipe s'instéresse principalement aux propriétés prouvables des algorithmes d'apprentissage automatique. Leurs domaines d'application couvrent la santé, la mobilité, les sciences sociales, les technologies vocales.

Les trois axes principaux des activités de recherche de l'équipe sont :

1. MAchine learninG in information NETworks

— Extraction et apprentissage dans les graphes
— Apprentissage automatique pour le traitement du langage naturel
— Apprentissage automatique décentralisé

On peut citer les derniers projets au sein de l'équipe :

— TRUMPET : TRUstworthy Multi-site Privacy Enhancing Technologies. 2022-2025
— PRIDE : Privacy-Preserving Decentralized Machine Learning ANR, 2020-2024
— PMR : Privacy-preserving methods for Medical Research, ANR, 2021-2025
— IMPRESS : Improving embeddings with semantic knowledge, Inria-DFKI 2020-2024
— SLANT : Spin and bias in language analyzed in news and texts, ANR PRCI 2020-2024

5.5 Stage proposé

Magnet développe une librairie d'apprentissage fédéré. Cette librairie sera rapidement mise en test au sein du groupement G4 d'hôpitaux de la région. L'équipe entretien également une collaboration avec l'équipe Epione qui développe une librairie similaire. L'objectif à moyen terme est de rapprocher ces deux librairies et l'effort de développement.

Le stage qui m'a été proposé fut de réaliser les tests définitifs pour la mise en test du code développé par Magnet, d'ajouter quelques fonctionnalités d'entrées-sorties qui pourront faciliter le déploiement et la récolte des résultats. Dans un deuxième temps, d'étudier et comparer la librairie avec celle de l'équipe Epione.

Le stage de 9 semaines s'est déroulé à l'INRIA, encadré par M. Marc Tommasi et Mme Nathalie Vauquier. L'environnement technique est python et git avec une bonne connaissance du shell.

5.6 Missions accomplies

Mes missions durant ce stage ont évoluées au fur et à mesure de ma prise en main avec le projet. En premier lieu, j'ai pu prendre connaissance de la documentation, comprendre le fonctionnement globale des grandes classes et des méthodes de communication entre clients et serveur.

Une fois familiarisé avec le projet, j'ai effectué une analyse de la couverture en test du projet avec pytest, puis proposé quelques tests de code non couvert par des tests.

Il m'a ensuite été demandé de proposer des améliorations des scripts bash. Ainsi après compréhension des scripts, j'ai pu comprendre les difficultés que posés les scripts pour l'équipe et après quelques échanges avec mon tuteur, j'ai pu être en capacité de proposer des solutions pour éviter le lancement d'expériences déjà effectuées, proposer une nouvelle organisation de la journalisation par session et par expériences, simplifier les scripts et les ouvrir à l'évolution du software avec l'aide d'un ingénieur de l'équipe. En fin de stage, j'ai pu acquérir les images des machines virtuelles des clients pour pouvoir déployer les scripts sur celles-ci et ainsi tester dans l'environnement le bon fonctionnement en production de la refonte des scripts.

Contribution

6.1 Introduction du projet : Declearn

> Avec l'apprentissage fédéré, l'innovation réside dans la création d'algorithmes d'apprentissage capables de fonctionner à partir de données stockées dans le réseau, sans avoir à les transmettre vers un lieu unique
>
> *Marc Tommasi*
> *responsable de Magnet*

L'équipe-projet a remporté en 2021 un appel à projets de la CNIL [1] pour « déployer, au sein d'un réseau de centres hospitaliers universitaires, des algorithmes de calcul fédéré dans le cadre d'études cliniques multicentriques dites décentralisées ». Le projet utilise les données de santé des patients des hôpitaux, cette collaboration impose donc un certain nombre de contraintes fortes pour garantir la sécurité et la confidentialité des données personnelles des patients des hôpitaux, qui sont des données sensibles soumises à des lois qui encadrent leur manipulation. L'apprentissage décentralisé prend ainsi sens dans ce contexte d'étude multicentrique , les centres hospitaliers avec lesquels Magnet collabore sont réunis au sein du Groupement de coopération sanitaire G4 (comprenant les CHU d'Amiens, de Caen, de Lille et de Rouen) et souhaitent éviter la centralisation pour garder le contrôle sur les données qu'ils collectent au travers de leurs activités.

6.1.1 Fonctionnement du modèle d'apprentissage fédéré

On retrouve un serveur qui ne possède pas de données, sur lequel nous allons nous appuyer pour orchestrer le calcul et les quatres hôpitaux du groupement G4 avec chacun sur leurs serveurs les données de santé de leurs patients. La fonction choisie dans le projet pour mesurer la qualité du modèle est une fonction de perte par moyenne. Ainsi nous pouvons illustrer le modèle d'apprentissage fédéré développé par l'équipe dans le contexte :

1. Commission nationale de l'informatique et des libertés

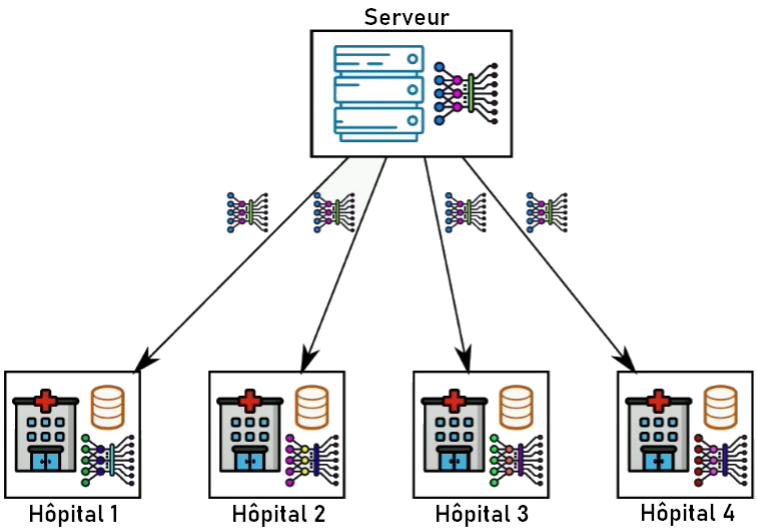

FIGURE 6.1 – Schéma modèle d'apprentissage fédéré G4

Source: Aurélien Bellet - Conférence Institut Henri Poincaré

Le serveur et les hôpitaux (*clients*) ne font donc pas tourner le même algorithme car le serveur maître lance un algorithme qui permet d'orchestrer les calculs. Voici un exemple des algorithmes utilisés côté serveur et client :

- Ensemble K contenant les clients
- Chaque client possède un dataset D_k
- On nomme θ les paramètres du modèle (ex les poids du réseau de neurones)
- Recherche des paramètres qui minimise l'erreur de prédiction globale : $min_\theta \sum_{k=1}^{K} Loss(\theta; D_k)$

Algorithm 1 FedAvg (server-side)

initialize θ
for each round $t = 0, 1, ...$ **do**
  for each party in k in parallel **do**
    $\theta_k \leftarrow ClientUpdate(k, \theta)$
  end for
  $\theta \leftarrow \sum_{k=1}^{K} \theta_k$
end for

Algorithm 2 ClientUpdate(k,θ)

Parameters : # steps L, step size η
for $1, ..., L$ **do**
  $\theta \leftarrow \theta - \eta \nabla Loss(\theta; D_k)$
end for
send θ to server

Une première étape d'initialisation est effectuée pour transmettre le modèle θ. Puis l'apprentissage se déroule en round comprenant du calcul local et de la communication, c'est-à-dire que chaque client va utiliser ses données pour faire une mise à jour de son modèle local pour ajuster celui-ci sur ses propres données (cf. Algorithm 2 ClientUpdate). Enfin, les clients communiquent leurs modèles mise à jour au serveur pour que ceux-ci soient agrégés au sein du serveur principal, puis celui-ci communique à son tour à chaque client le modèle global qui en résulte puis on recommence avec les évolutions du modèle.

6.1.2 Communication et contraintes

On peut constanter que ce système nécessiste une couche de communication entre le serveur et les clients. Plusieurs contraintes sont posés par la commission des hôpitaux :

— Respect des pare-feu des hôpitaux
— Interdiction d'utiliser le protocole SSH
— Ce sont les hôpitaux qui se connectent au serveur non le serveur qui se connecte aux clients

Pour répondre à ces contraintes l'équipe a développé le projet dans un premier temps avec le protocole websocket puis gRPC qui est devenu le protocole principal du projet. Le protocole gRPC [2] qui utilise le protocole HTTP/2 permet un respect des pare-feu des hôpitaux ainsi que la contrainte sur le protocole SSH. Puis pour respect le sens de connexion, le serveur sera démarré en premier puis aura une socket en écoute en l'attente d'un nombre de clients (les hôpitaux), ainsi les clients se connecteront sur l'adresse ip et le port du serveur que l'aura définit au préalable. Cette manœuvre est notamment gérer par les scripts de production.

6.1.3 Structure de Declearn

Voici le schéma de dépendance et de la structure par classe de Declearn :

2. framework RPC open source initialement développé par Google

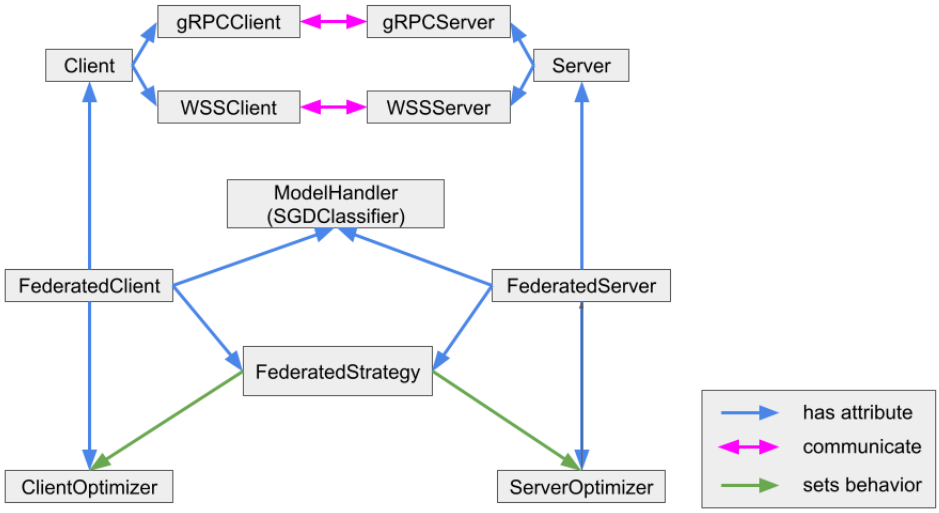

FIGURE 6.2 – Schéma de dépendance et communication de Declearn

Source: Documentation du projet Declearn - Magnet INRIA

— FederatedServer et FederatedClient sont respectivement les classes d'encapsulation pour le serveur et les clients.
— Le serveur et le client sont les objets utilisés pour communiquer pendant l'apprentissage du modèle fédéré.
— FederatedStrategy est l'objet qui définit les deux hyperparamètres FL (nombre de tours, seuil à atteindre, taille du lot, comportement des taux d'apprentissage du serveur et du client, etc.) et algorithme (FedAvg, etc.). Étant instancié au niveau du serveur, il construit le ServerOptimizer en fonction des paramètres reçus et il construit le message envoyé au FederatedClient qui sera utilisé pour construire le ClientOptimizer
— ServerOptimizer exécute l'agrégation côté serveur, son comportement dépend de la stratégie choisie, tandis que ClientOptimizer exécute les mises à jour locales côté client, son comportement dépend également de la stratégie choisie.

6.1.4 Chaîne de production du modèle

6.1.4.1 Planificateur de tâches et scripts bash

Pour respecter la contrainte du sens de connexion entre le serveur et les hôpitaux, l'équipe utilise un planificateur de tâches (une crontab) pour lancer l'exécution de scripts bash qui lanceront le programme python soit du serveur soit du client et ainsi démarrer une session dans laquelle est exécutée une expérience avec certains paramètres donnés au lancement du programme et d'autres communiqués par le serveur. Une fois toutes les expériences lancées et donc la session terminée, la journalisation est envoyée sur un dépôt git à l'aide d'un script par utilisation du protocole HTTP.

Ainsi le projet est stockée sur un dépôt git contenant tous les fichiers du programme python ainsi que les scripts bash de production. Ce dépôt est cloné sur le serveur et sur les machines des hôpitaux, on y retrouve notamment des fichiers de configuration contenant certains paramètres, les informations concernant le dépôt git et les chemins absolues vers les fichiers d'expériences. La crontab du serveur et des clients va lancer à intervalle fixe (ex chaque demi-heure) le script de

production, celui-ci effectuera si nécessaire un *git pull* c'est-à-dire une mise à jour du dépôt local si des modifications ou ajouts sont apportés au dépôt git en ligne du projet.

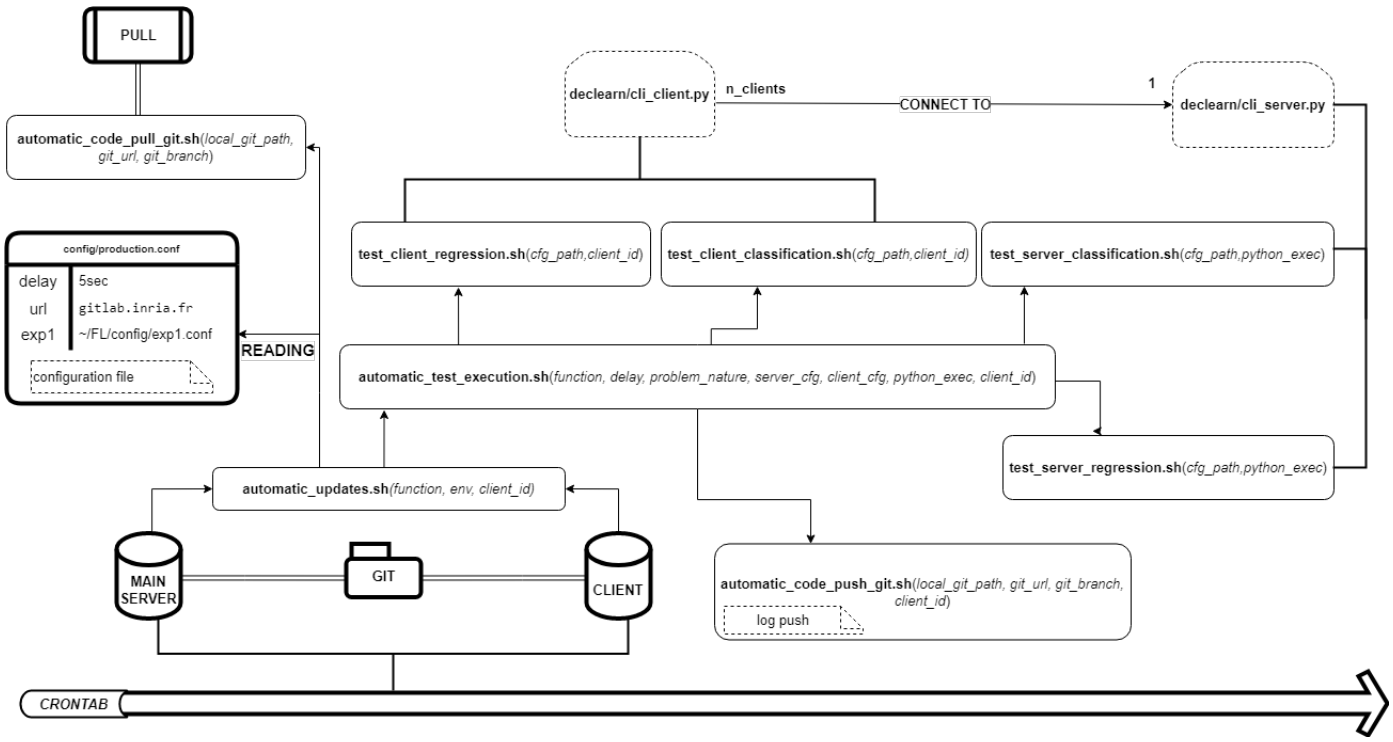

FIGURE 6.3 – Diagramme UML des scripts bash de production

La crontab est configurée de façon à ce que le serveur et les machines des hôpitaux exécutent en même temps le script principal *automatic_updates.sh*. Ce script exécute dans l'ordre :

- la récupération des paramètres et expériences dans le fichier de configuration, puis un *git pull* si nécessaire
- toutes les expériences du fichier de configuration à l'aide du script *automatic_test_execution.sh* qui en fonction de la nature du problème pour l'expérience (régression ou classification) et en fonction de la machine (serveur ou client/hôpital) utilisera les scripts *test_client_regression.sh* et $test\_client\_classification.sh$ ainsi que $test\_server\_regression.sh$ et $test\_server\_classification.sh$
- Un délai de lancement de script est effectué pour que le script qui démarrera le serveur sera exécuté avant les scripts qui démarrerons les n clients
- Le lancement du serveur par utilisation du programme python *cli_server.py* ou le lancement des clients par utilisation du programme python *cli_client.py*
- Pour chaque expérience, le serveur attend que tous les clients soient connectés à celui-ci avant de lancer une expérience
- Enfin, une fois toutes les expériences lancées, la journalisation de chaque client et du serveur sont envoyés sur le dépôt git dédie aux logs avec le script *automatic_code_push_git.sh* qui utilise la commande *git push* sous protocole HTTP.

6.1.4.2 Inconvénients

Cette production du modèle ne convient pas à l'équipe pour plusieurs raisons. Premièrement, les scripts sont en trop grand nombre ce qui rend la compréhension de leurs fonctionnement complexe aussi bien pour l'équipe que pour les ingénieurs des hôpitaux. Deuxièmement, l'implémentation faite des expériences au niveau des scripts fait que le planificateur de tâches relance à intervalle de temps fixe les scripts de production et donc toutes les expériences mentionnées dans le fichier de configuration de production des scripts. L'intervention de l'équipe est donc nécessaire pour retirer les expériences qui se sont bien déroulées lors de la dernière exécution des scripts, auquel cas les mêmes expériences sont lancées à chaque intervalle de la crontab.

L'implémentation des scripts de production qui va être présentée constitue ma contribution au projet Declearn pour répondre aux inconvénients ci-dessus.

6.2 Amélioration des entrées-sorties du framework

L'amélioration attendue des entrées-sorties du framework me fut présenté dès ma première semaine de stage, celle-ci consiste en une refonte des scripts pour répondre aux inconvénients que ceux-ci posent à l'équipe et est très attendue par l'équipe et par les ingénieurs des hôpitaux. Il m'a donc été demandé de réfléchir à des solutions, puis de les présenter au directeur de l'équipe pour vérifier la conformité de celles-ci.

6.2.1 Gestion des expériences

Le problème concernant les expériences est que les expériences déjà effectuées au n-ième lancement du premier script par la contrab, le sont de nouveau au lancement du premier script à la $n+1$-ième. Ce qui n'hésite donc l'intervention d'un membre de l'équipe pour mettre à jour le fichier de configuration sur le dépôt git et ainsi retirer les expériences qu'ils ne souhaitent pas relancer du fichier qui sera mise à jour dans les machines au prochain lancement des scripts par la crontab.

La solution proposée est d'utiliser une table d'expériences qui contiendra les couples $(id, path)$, où l'id est un identifiant unique attribué par ordre croissant sur la table avec pour premier id la valeur 1, et $path$ le chemin vers le fichier de configuration de l'expérience qui contient les informations suivantes et nécessaire pour une expérience :

— name : nom de l'expérience
— model : nature du problème (Regression : sgdegressor, Classification : sgdclassifier)
— n_participants : nombre de clients/hôpitaux
— epoch : nombre de passage sur le batch
— rounds : nombre maximum de rounds (cf. Algorithm 1 FedAvg)
— batch_size : taille du batch

Cette table sera stockée dans le fichier $config/.todo$ qui sera un fichier suivi, c'est-à-dire qu'il sera pris en compte dans la mise à jour du dépôt git, donc concerné par toutes modifications et par les commandes *git push* et *git pull*. Voici un exemple d'une telle table :

1 config/expeHeart1.conf
2 config/expeHouse.conf

3 config/expeHeart2.conf
4 config/expeHouse.conf

Ensuite, nous nous appuyerons sur un fichier $config/.done$, qui est local et propre à chaque client, c'est-à-dire qu'il ne sera pas suivi et donc concerné par aucune manipulation avec le dépôt git. Pour garantir ceci, le fichier est ajouté dans la liste des fichiers ignorés du dépôt git. Chaque client aura au préalable lors du déploiement, ce fichier avec la valeur 0 écrite dans le fichier. Cette valeur représente l'identifiant (id) de la dernière expérience lancée et terminée sans erreur. Ce qui signifie que si un code d'erreur est retourné par le script *start_server.sh* ou *start_client.sh* alors le fichier *.done* local ne sera pas édité pour y écrire l'identifiant de l'expérience qui s'est déroulée et l'exécution des scripts prendra fin dès réception du code d'erreur. Ainsi si il y a deux expériences à lancer et que la première génère une erreur alors le(s) script(s) en cours d'exécution sont arrêtés et donc la deuxième expérience n'est pas lancée. Nous fournissons également un script *clean_all.sh* qui à son exécution remplacera la valeur écrite dans $config/.done$ par la valeur 0.

Après divers échanges et modifications avec le directeur de l'équipe, nous avons convenu de cette solution pour répondre aux problèmes liés aux expériences dans la chaîne de production de Declearn.

6.2.2 Organisation et envoi de la journalisation

La journalisation du projet est effectué par les scripts avec une redirection vers un fichier de log de toute la trace d'exécution des scripts et exécutions python. Puis à la fin du script $automatic_updates.sh$, le script $automatic_code_push_git.sh$ est utilisé pour envoyer sur le dépôt git dédié à la journalisation tous les fichiers de résultats et journalisation.

L'alternative proposée dans cette refonte des scripst est de toujours utiliser des redirections vers des fichiers de log de la trace d'exécution des scripts et des exécutions python. Mais en apportant une arborescence aux fichiers pour les ordonner par session de crontab et expériences. Par exemple si je planifie un lancement d'une session avec la crontab le mercredi 20 juillet 2022 à 14h21 alors nous obtiendrons l'arborescence suivante :

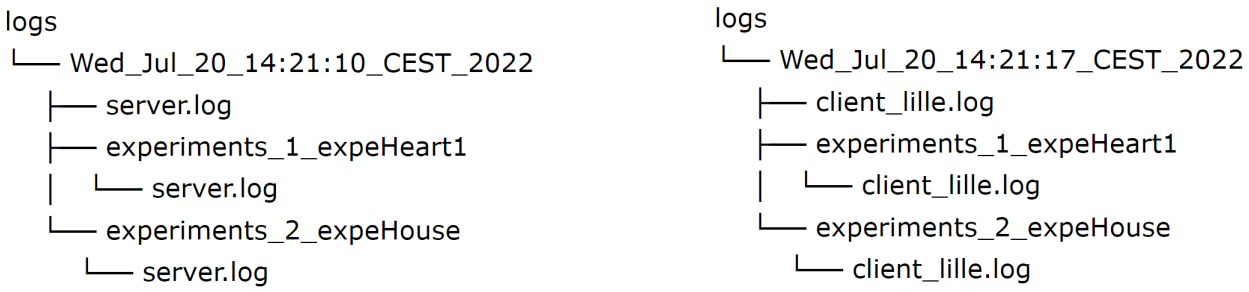

FIGURE 6.4 – Exemple d'arborescence du dossier de journalisation

Ainsi on retrouve le dossier horodaté de la session de crontab, où se trouve les fichiers *.log* du serveur et des clients qui contiennent les traces d'exécutions des scripts et des exécutions python. Puis on y retrouve des dossiers qui portent le nom de l'expérience ainsi que l'identifiant de

l'expérience dans la table au moment de la session, et dans ces dossiers on y retrouve des fichiers *.log* pour le serveur et les clients qui contiennent uniquement les traces d'exécutions python du software.

Cette nouvelle organisation de la journalisation permet de consulter plus facilement et de façon plus ciblée un fichier de log, ce qui facilite la lecture des logs en cas de débogage et permet de mieux reproduire le contexte de déclenchement d'erreurs pour tester les modifiations apportées pour résoudre les erreurs.

Concernant l'envoi des fichiers de logs, l'utilisation d'un dépôt git permet de respecter la contrainte au niveau du protocole utilisé, mais ne consitute pas une solution élégante sachant qu'une solution réseau est implémenté directement dans le projet. La solution proposée est de créé un programme python qui permet la compression et l'envoi par le canal de communication créé et utilisé entre le client et le serveur lors d'une expérience pour envoyer au serveur le dossier de logs de l'expérience de chaque client qui vient d'être effectuée. Hors dans le cas où il y aurait une erreur réseau par exemple alors cette transmission du dossier ne serait possible. Ainsi nous proposons un script *sendlogs.sh* qui pourra être lancé en fin de session pour indiquer aux machines des clients de lancer le programme python créé pour l'envoi des logs, qui effectuera une compression du dossier de logs complet ce qui comprend donc toutes les sessions, toutes les expériences et fichiers de logs présent l'arborescence du dossier *logs* (cf. Figure 5.4).

Ma contribution à ce programme python pour l'envoi des dossiers et fichiers de logs s'est limité à une aide dans la compréhension et approche de construction de celui-ci auprès de mon binôme stage qui eut pour mission de faire ce programme python qui est donc complémentaire à ma contribution sur la nouvelle gestion de la journalisation qui nécessiste donc une nouvelle méthode d'envoi.

6.2.3 Nouvelle structure de la chaîne de production

Ces améliorations ont nécessité une refonte complète des scripts pour pouvoir fournir une structure simple d'utilisation et de compréhension tout en répondant aux attentes de l'équipe. Voici le diagramme de la nouvelle chaîne de production de Declearn :

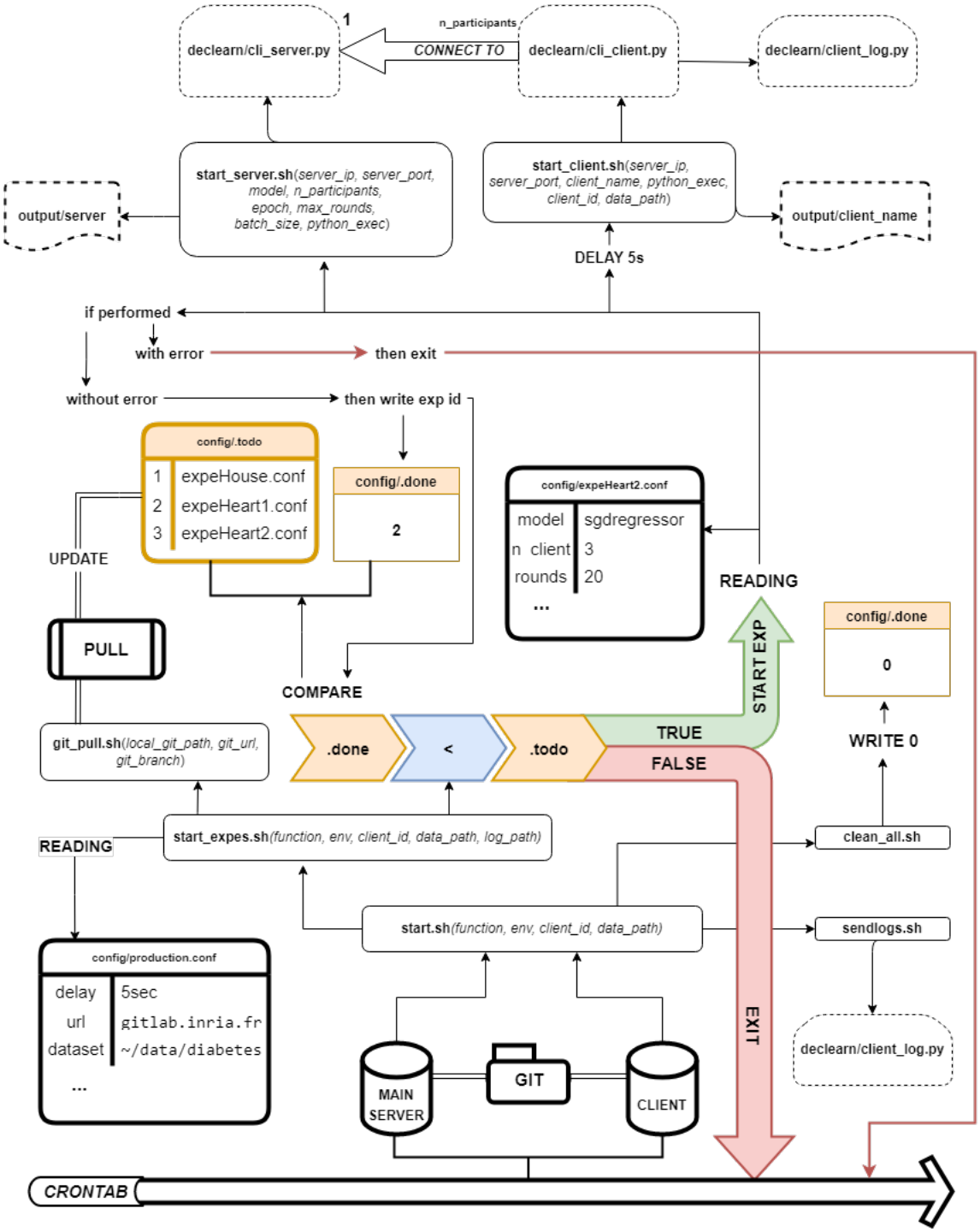

FIGURE 6.5 – Diagramme UML de la refonte des scripts bash de production

Avec cette nouvelle implémentation des scripts, il suffit alors à l'opérateur d'ajouter des expériences par couple (*id*, *path*) dans le fichier $config/.todo$ et d'effectuer un *git push*. Ainsi la crontab, exécutera le script *start.sh* avec les options nécessaire, par exemple :

Crontab serveur :

```
$ crontab -e
0,30 * * * * bash /home/server/FL/production/start.sh -f SERVER
                                                        -t PRODUCTION
```

Crontab client :

```
$ crontab -e
0,30 * * * * bash /home/client/FL/production/start.sh -f CLIENT
                                                    -t PRODUCTION -i ~/.client_id
```

Le script *start_expes.sh* sera alors exécuté par le script *start.sh*, avec les options qui lui sont donnés et les redirections et chemin vers les dossiers et fichiers de journalisation. Le script *start_expes.sh* vérifie si le dépôt local du projet est à jour par rapport au dépôt git du projet, si nécessaire il effectue alors un *pull* qui mets à jour les modifications apportées dont le fichier *config/.todo*. Si le dernier identifiant de la table d'expérience est plus grand que l'identifiant stocké localement dans le fichier non suivi *config/.done* alors la liste des expériences à effectuées sont lancées les unes à la suite des autres si aucune erreur ne se produit auquel cas un arrêt de la chaîne est effectué. Chaque expérience possède maintenant son propre fichier de configuration comportant ces propres paramètes. Le serveur est lancé à l'aide du script *start_server.sh* et les clients à l'aide du script *start_client.sh* qui est lancé avec un délai en secondes précisé dans le fichier de configuration *config/production.conf* qui est commun au serveur et aux clients. Si une expérience se termine sans erreur alors avant la fermeture du canal de communication établi entre le client et le serveur, la journalisation de l'expérience est transmise par à l'aide du programme *declearn/client_log.py*, cette transmission ne peut avoir lieu si il y a une erreur lors de l'expérience ou un soucis réseau.

Puis, après avoir terminé l'exécution du script *start_expes.sh*, le script *start.sh* est en capacité d'exécuter les scripts *clean_all.sh* pour réinitialiser la valeur dans le fichier *config/.done* à 0. Comme nous ne souhaitons pas réinitialiser cette valeur à chaque exécution du script *start.sh*, la ligne spécifique à cette exécution est commenté, il suffit de la décommenter pour que celui-ci l'exécute et d'effectuer un *push* pour que tous les clients prennent en compte ceci, sinon un client peut décommenter localement ou lancer le script sur sa machine.

Enfin, de la manière que pour le script *clean_all.sh*, l'exécution du script *sendlogs.sh* est possible en fin du script *start.sh* si l'on décommente la ligne. Ce qui exécutera l'envoi complet du dossier de journalisation de chaque client au serveur à l'aide du programme *declearn/client_log.py*, si la ligne est décommentée et la modification ajouté sur le dépôt git avec un *push*, sinon un client peut décommenter localement ou lancer le script sur sa machine pour effectuer l'envoi sans attendre la crontab.

Développement personnel et professionnel

7.1 Enjeux environnementaux du numérique

Durant l'année, il m'a été permis de participer à un atelier pour comprendre en équipe et de manière ludique les enjeux environnementaux du numérique. Cet atelier était autour de la fresque du numérique qui est similaire à celle de la fresque du climat.

J'ai notamment été marqué par l'impact de l'industrie minière sur l'environnement ou sur les tensions géopolitique provoquées par celle-ci dans les régions d'exploitation. Les chiffres clés donnés durant l'atelier m'a permis de me rendre compte de ces aspects sur plusieurs niveaux d'échelles. Les chiffres donnés durant l'atelier permettent également de remettre en question concernant notre confort numérique au vue des enjeux environnementaux.

Suite à cet atelier, j'ai souhaité approfondir le sujet en effectuant des recherches personelles. J'ai pu découvrir au travers de mes recherches l'association SystExt (Systèmes extractifs et Environnements) qui regroupe des professionnels en activité qui ont un intérêt commun pour les systèmes extractifs, en particulier miniers, et des compétences pour répondre aux problématiques techniques associées à ces activités. Cette association est constituée d'ingénieurs géologues et miniers, mais également de membres d'autres disciplines des sciences et politiques environnementales. J'y ai découvert notamment Mme Aurore STEPHANT, ingénieure géologue minier spécialisée dans les risques environnementaux et sanitaires des filières minérales, qui donne des interviews et conférences sur l'impact de l'industrie minière sur notre environnement. Et qui nous permet de nous avertir sur notre modèle de société ou sur des problèmatiques en cas de pénurie de métaux et métaux précieux pour le secteur du numérique.

Durant mon stage, j'ai pu discuter de la question de l'impact sur l'environnement du numérique avec mon tuteur de stage, le directeur de l'équipe Magnet M. Marc TOMMASI. Il m'a indiqué qu'il est personnellement sensibilisé par la question, il est notamment animateur durant les ateliers de la fresques du numérique à l'université de Lille. Il est également membre de la commission du développement durable de l'inria. Mais se rend aussi à des comités scientifiques nationaux durant lesquels des exposés et conférences de sensibilisation sont donnés sur la question de l'impact du numérique. Il y a notamment des membres du GIEC qui y sont présents. Au titre de chercheur, M. Tommasi m'a également appris que les projets scientifiques sont soumis à des points de validation sur leur impact énergétique. Et qu'un devoir d'autocritque au niveau éthique et environnemental de son travail est demandé lors de la soumission de celui-ci à la communauté scientifique pour pu-

blication.

La commission locale du développement durable au sein de l'inria s'intéresse à la question de l'impact du numérique notamment de la consommation énergétique des batîments de l'inria. L'inria cherche également à réduire ces déchets, pour cela par exemple plus de poubelles sont mises à disposition dans les bureaux mais également des bacs de tri dans les couloirs. Curieusement, Une réduction de la récolte des déchets fut observable après ces initiatives. Au niveau de l'équipe Magnet, des choses sont mises en place pour limiter le plus possible l'impact des travaux de l'équipe sur l'environnement. Cela passe par la mesure de leur impact et le contrôle de celui-ci dans leur projet ainsi que la prise en compte de la consommation en énergie des algorithmes d'apprentissage. Ou encore en limitant les achats de machines ou matériels informatiques.

Avec l'aide de M. Tommasi, j'ai notamment pu me documenter sur l'impact de l'apprentissage machine sur l'environnement, mais également de l'apprentissage centralisé, décentralisé et leur comparaison. La question de l'apprentissage fédéré pour réduire l'impact sur l'environnement de l'apprentissage machine est actuellement toujours ouverte et constitue la problèmatique de nombreuses recherches dans la littérature. Effectivement l'apprentissage fédéré commence à être déployé à l'échelle mondiale par des entreprises qui doivent adhérer à de nouvelles demandes juridiques et politiques émanant des gouvernements et de la société civile pour la protection de la vie privée. Cependant, l'impact sur l'environnemental potentiel lié à ce secteur reste vague et inexploré. Une étude de 2020, "Can Federated Learning Save The Planet ?" montre que l'apprentissage fédéré peut être une technologie plus verte que les GPU des data center.

Conclusion

Je retiens une note positive, en tout point, de mon stage de fin d'études dans le cadre de la Licence 3 Informatique parcours Info. J'ai eu l'honneur d'être intégré à une équipe de recherche. Cette expérience m'a permis de me faire une idée concrète des exigences d'un poste au sein d'une équipe de recherche mais également conforté sur l'importance de la recherche publique dans notre société, pour la société.

La formation acquise durant mon cursus à l'université de Lille m'a été utile dans différentes situations, que ce soit sur le plan technique (programmation, maintenance système et réseau, bonnes pratiques) que sur le plan relationnel (travail collaboratif, gestion du temps, présentation du travail, etc.). Durant ce stage, j'ai pu perfectionné mes connaissances en bash, shell, programmation objet mais également à améliorer mes méthodes de travail et de présentation de mon travail pour faire comprendre mes idées.

Grâce à l'appui de membres et du directeur de l'équipe, j'ai pu être en capacité de proposer des solutions pour répondre aux problèmes majeurs des entrées-sorties en restructurant la chaîne de production, en apportant une nouvelle organisation de la journalisation et en prenant part aux idées à développer concernant une nouvelle méthode de transmission des fichiers de journalisation.

Les nouvelles fonctionnalités développées et les modifications apportées, vont permettre à l'équipe Magnet de pouvoir mieux manipuler la chaîne de production, de faire évoluer leur projet plus facilement, d'offrir aux hôpitaux du groupement de coopération sanitaire G4 une mise en production moins complexe et nécessister moins d'intervention des ingénieurs des hôpitaux sur la chaîne de production.

Ce stage se termine avec la réalisation de la mission prioritaire que le directeur de l'équipe a fixé : l'amélioration des entrées-sorties du framework que constitue la refonte des scripts et la nouvelle gestion de la journalisation. Cependant certains objectifs cités au départ comme l'ajout de tests à la couverture du projet ou la comparaison avec la librairie de l'équipe Epione ne fut possible sur les neuf semaines de stage.

Bilan

Cette expérience professionnelle m'a permis de découvrir un peu mieux le monde de la recherche, et de me faire une idée plus concrète des travaux effectués au sein d'un laboratoire de recherche en informatique. Profesionnellement, je sors plus autonome, proactif et rigoureux dans mon travail.

Plus personnellement, j'ai commencé ce stage avec un certains nombre de questions mais en sors avec autant, tant les thématiques traitées par l'équipe Magnet sont inspirantes et de mon point de vue importantes dans le contexte de l'évolution du domaine informatique et de la place de celui-ci dans notre société, notamment sur les questions de confidentialité et souveraineté de nos données personnelles.

Durant ma formation à l'université de Lille, j'ai pu acquérir un ensemble de savoir, de connaissances dans le domaine informatique qui m'ont permis de mêner à bien ce stage et d'être armé dans des situations de difficultés techniques mais j'ai également pu apprendre beaucoup de choses en adéquation avec ma formation au sein de l'équipe Magnet. Enfin, Ce stage rentre en cohérence avec mon projet d'étude à la rentrée prochaine en Master Machine Learning. Mon projet professionnel se porte plus sur le secteur de la recherche mais je reste ouvert et également prudent au vue des compétences nécessaire pour exercer dans ce domaine.

Bibliographie

[1] Inria article (2022) : Magnet prescrit l'apprentissage fédéré aux établissements de santé.

[2] Aurélien Bellet (2021) Decentralized and Privacy-Preserving Machine Learning, conférence à l'Institut Henri Poincaré

[3] arXiv :2010.06537, Xinchi Qiu, Titouan Parcollet, Daniel J. Beutel, Taner Topal, Akhil Mathur, Nicholas D. Lane, Can Federated Learning Save The Planet ?

Table des figures

6.1 Schéma modèle d'apprentissage fédéré G4 9
6.2 Schéma de dépendance et communication de Declearn 11
6.3 Diagramme UML des scripts bash de production 12
6.4 Exemple d'arborescence du dossier de journalisation 14
6.5 Diagramme UML de la refonte des scripts bash de production 16

List of Algorithms

1 FedAvg (server-side) 9
2 ClientUpdate(k,θ) 9