Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

Kasdi Merbah University - Ouargla

Faculty of New Information and Communication Technologies

FNTIC

Department of Computer Science and Information Technology

Academic Master Degree

Field: Mathematics and Computer Science

Track: Computer Science / Specialization: Fundamentals of Computer Science

Privacy-preserving techniques in FL.

Realized by : *Chetti Rayan*

Supervised by :

Dr. Akram Zine Eddine Boukhamla (UKMO)

Member of the jury:

Dr. Khaldi Amine: UKMO - President

Dr. Eushi Salah : UKMO - Examiner

Juin 2023

Thanks

I would like to express my gratitude to the entire teaching team for their availability and the quality guidance they provided during the preparation of my project. I would particularly like to thank my family for their support and the help they gave me, which was instrumental in completing this work.

ملخص

التعلم الموحد (FL) لديه القدرة على تدريب النماذج على البيانات اللامركزية أثناء الحفاظ على خصوصية البيانات. ومع ذلك، تعد الثقة والأمن من الاهتمامات الرئيسية في هندسة هذا النظام نظرًا لاحتمال وجود مساهمين ضارين. يهدف هذا العمل إلى تحسين دیمومة نظام التعلم الموحد من خلال معالجة التخفيف القائم على الثقة في النظام المذكور. تستكشف الدراسة أيضا أنواعًا مختلفة من الثقة ، بما في ذلك جدارة المساهمين بالثقة ، موثوقية النموذج وتحيز البيانات. تقترح هذه الورقة إطار التخفيف مبني على الثقة متعدد الأوجه لنظام التعلم الموحد ، بما في ذلك النمذجة القائمة على السمعة ،بروتوكولات التجميع الآمن وتقنيات الحفاظ على الخصوصية. هذه الآليات تتعرف وتعالج المشاركين غير الموثوق بهم أو سيئي النية، مما يضمن سلامة نظام التعلم الموحد. كما يتتبع آثار تقنيات التخفيف المبنية على الثقة على الأداء وكفاءة أنظمة التعلم الموحد، موازنة التدابير الأمنية والنفقات الحسابية. يتم اختبار إطار العمل من خلال التجارب والمحاكاة باستخدام مجموعات البيانات الواقعية والسيناريوهات وتقدير أدائها من حيث دقة النموذج ومعدل التقارب ، فعالية الاتصال والتشابه مع سيناريوهات التطبيق في العالم الحقيقي ضد الهجمات العدائية. تساهم هذه الدراسة في مجال التعلم الموحد من خلال معالجة مسائل الثقة التحديات وتوفير استراتيجيات التخفيف الفعالة ، مما يمهد السبل لمزيد من الأمن ونظام تعلم موحد جدير بالثقة في المجالات الحساسة للخصوصية مثل الرعاية الصحية والتمويل ، والمدن الذكية.

الكلمات المفتاحية: التعلم الموحد ، بروتوكولات التجميع الآمن ، تقنيات التخفيف المعتمدة على الثقة ، المحاكاة ، نظام التعلم الموحد ، التعلم الآلي.

Abstract

Federated Learning (FL) has the potential to train models on decentralized data while maintaining data privacy. However, trust and security are major concerns in federated learning architecture due to the possibility of malicious contributors. This essay aims to improve the longevity of the federated learning system by addressing trust-based mitigation. The study explores different types of trust, including contributor trustworthiness, model trustworthiness and data bias. In addition, it examines trust-based mitigation techniques for the FL system, including reputation-based modelling, secure aggregation protocols and privacy-preserving techniques. These mechanisms recognise and deal with the non-trusted participants, ensuring the integrity of the FL system. The study also investigates the effects of trust-based mitigation techniques on the performance and efficiency of FL systems, balancing security measures and computational load. The approach is tested through experiments and simulations using real-world datasets and scenarios, estimating its performance in terms of model accuracy, convergence rate, communication efficiency and resemblance to application scenarios against adversarial attacks. This essay contributes to the field of FL by addressing trust challenges and providing effective mitigation strategies, paving the way for a more secure and reliable FL system in sensitive and private domains such as healthcare, finance and smart cities.

Keywords

FL system, trust , security, privacy, Secure aggregation protocols, Trust-based mitigation techniques, Simulation , , Machine learning.

Résumé

L'apprentissage fédéré (Federated Learning) a le potentiel de former des modèles sur des données décentralisées tout en maintenant la confidentialité des données. Cependant, la confiance et la sécurité sont des préoccupations majeures dans l'architecture d'apprentissage fédéré en raison de la possibilité de contributeurs malveillants. Cet essai vise à améliorer la longévité du système d'apprentissage fédéré en abordant l'atténuation basée sur la confiance. L'étude explore différents types de confiance, notamment la fiabilité des contributeurs, la fiabilité des modèles et la partialité des données. En outre, elle examine les techniques d'atténuation basées sur la confiance pour le système d'apprentissage fédéré, y compris la modélisation basée sur la réputation, les protocoles d'agrégation sécurisés et les techniques de préservation de la vie privée. Ces mécanismes reconnaissent et traitent les participants non fiables, garantissant ainsi l'intégrité du système FL. L'étude examine également les effets des techniques d'atténuation basées sur la confiance sur les performances et l'efficacité des systèmes de FL, en équilibrant les mesures de sécurité et la charge de calcul. L'approche est testée par le biais d'expériences et de simulations utilisant des ensembles de données et des scénarios du monde réel, estimant ses performances en termes de précision du modèle, de taux de convergence, d'efficacité de la communication et de ressemblance avec les scénarios d'application contre les attaques adverses. Cet essai contribue au domaine du FL en abordant les défis de confiance et en fournissant des stratégies d'atténuation efficaces, ouvrant la voie à un système FL plus sûr et plus fiable dans des domaines sensibles et privés tels que les soins de santé, la finance et les villes intelligentes.

Mots clés

système FL, confiance, sécurité, la vie privée, protocoles d'agrégation sécurisés, techniques d'atténuation basées sur la confiance, simulation.

Table des matières

Table des figures

Abbreviations

AI Artificial Intelligence

FL Federated Learning

IoT Internet of Things

P2P Peer-to-Peer

ML Machine Learning

DL Deep Learning

DP Differential Privacy

MPC Multi-Party Computation

SMC Secure Multi-Party Computation

GDPR General Data Protection Regulation

IID Independent and Identically Distributed

E2E End-to-End

FATE Federated AI Technology Enabler

TFF TensorFlow Federated

NLP Natural Language Processing

General Introduction

The year 2017 marked a significant turning point for Artificial Intelligence (AI) as AlphaGo Zero, developed by DeepMind, defeated professional chess players, demonstrating the immense potential of (AI). Since then, there has been a growing expectation for more advanced AI technologies to be implemented in various applications such as driverless cars, medical care, and finance. As a result, AI has shown its strengths in recent years across almost every industry and aspect of life. Nevertheless, still, the development of AI has been facing challenges and setbacks. One of the key drivers of the current public interest in AI is the availability of big data. AlphaGo Zero, for exam-ple, used 28.6 billion sets of human and machine-generated chess data for training to achieve remarkable re-sults. This success has led to high expectations that AI-driven by big data, similar to AlphaGo, will soon perme-ate every aspect of our lives. However, the reality could be better, as most industries need more access to high-quality data, making the application of AI technology more challenging than anticipated. For example, is it pos-sible to gather data from different sources and create a central data repository ? Unfortunately, breaking the barriers between data sources is difficult and impossible in many situations. Data for AI projects often come in multiple types and are scattered across isolated islands. Industries face competi-tion, privacy and security concerns, and complex administrative procedures, making data integration even within different departments of the same company a daunting task. Integrating data from multiple organiza-tions or across geographical regions is nearly impossible or prohibitively expensive. Additionally, as large companies compromise data security and user privacy, the importance of data privacy and security has become a primary global concern. Incidents of data breaches, such as the one involving Face-book, have raised significant public and government scrutiny. In response, Governments worldwide are enacting stricter data security and privacy laws. For example, the General Data Protection Regulation (GDPR), implemented by the European Union, aims to protect users' privacy and ensure data security. It requires businesses to use clear and transparent language in user agreements and grants users the right to have their data deleted or withdrawn, commonly known as the "right to be forgotten." In addition, non-compliant companies face substantial fines. Similar acts

and regulations are being enacted in the United States and China, emphasizing the need for data protection and privacy. These new regulations challenge the traditional data transaction procedures used in AI, which typically involve one party collecting and transferring data to another party for cleaning, fusion, and modelling. The resulting models are often sold as services. However, these transaction models may violate privacy and security laws, such as the GDPR, as users may need a clearer understanding of the future uses of their data. This essay presents an alternative approach called Federated Learning (FL) with differential privacy, which offers a possible solution to these challenges. We provide an overview of existing research on FL, define its characteristics, propose categorizations, and explore its applications within a comprehensive and secure federated learning framework. Finally, we examine how this framework can be successfully applied to various industries. By promoting federated learning, we aim to shift the focus of AI development from solely improving model performance to investigating methods that comply with data privacy and security laws for data integration.

Chapitre 1

OVERVIEW OF FEDERATED LEARNING

1.1 Machine Learning

According to [7] machine learning can be classified into four types :

a) **Supervised Learning :**

In supervised learning, the algorithm learns from labelled examples. The training data consists of input-output pairs, where the desired output is known. The algorithm then learns to map inputs to outputs based on these examples. For example, in a spam email classification system, the algorithm is trained on a dataset of emails labelled as spam or not spam, and it learns to classify new emails based on the patterns it discovers in the training data.

b) **Unsupervised Learning :**

Unsupervised learning involves learning from unlabelled data. The algorithm is tasked with finding patterns or structures in the data without any predefined output labels. Clustering is a common unsupervised learning technique where the algorithm groups similar data points together based on their inherent similarities or distances. Unsupervised learning is useful for tasks such as customer segmentation, anomaly detection, or dimensionality reduction

c) **Semi-supervised learning :**

Semi-supervised learning combines the benefits of both supervised and unsupervised learning. It leverages the labeled examples to guide the learning process, while also utilizing the unlabeled examples to extract additional information or patterns from the data.

d) **Reinforcement Learning :** Reinforcement learning involves an agent learning to interact with an environment to maximize rewards or minimize penalties. The agent learns by trial and error, receiving feedback in the form of rewards or punishments for its actions. Through a process of exploration and exploitation, the agent learns to take actions that lead to favourable outcomes. Reinforcement learning has been successful in applications such as game playing (e.g., AlphaGo) and robotic control.

In addition to these types, there are other specialized areas of Machine Learning (ML), such as *Deep Learning (DL)*, which is a subset of neural networks that involves training deep architectures with multiple layers to learn hierarchical representations of data. Deep learning has achieved remarkable success in various domains, including computer vision, natural language processing, and speech recognition. Machine learning techniques have revolutionized numerous industries and applications. They have been used for image and speech recognition, recommendation systems, fraud detection, autonomous vehicles, medical diagnosis, and much more. The ability to extract insights and patterns from large volumes of data has opened new possibilities and improved decision-making processes across various domains.

1.2 Federated Learning

Federated learning is a novel machine learning technique that facilitates collaborative model training across numerous decentralized or edge devices, while ensuring data privacy and security. It tackles the issues associated with conventional centralized machine learning, where data is usually gathered and saved in a central server. Federated Learning is a distributed learning paradigm with two key challenges that differentiate it from traditional distributed optimization :

— significant variability in terms of the systems characteristics on each device in the network (systems heterogeneity).

— non-identically distributed data across the network (statistical heterogeneity).

Below is a summary of the federated learning procedure :

1. **Initialization and Setup :**

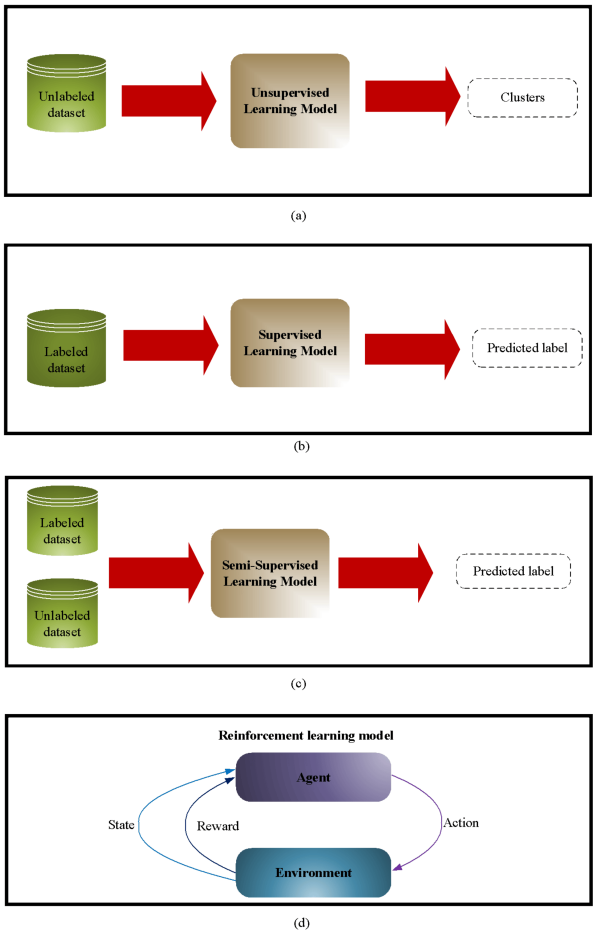

FIGURE 1.1 – Different learning schemes : (a) Unsupervised learning, (b) Supervised learning, (c) Semi-supervised learning, (d) Reinforcement learning.[14]

— Device Selection : A group of devices is selected to take part in the federated learning procedure. These devices may belong to individuals, institutions, or be dispersed across various geographic regions.

— Model Initialization : The central server starts a global model or a set of initial model parameters. This model serves as a foundation for the collaborative training process.

2. **Local Model Training :**

   Each device trains its own model using its local data, without sharing it with the central server or other devices. During the training process, the device computes gradients based on its local data and updates its local model parameters, accordingly, representing the direction of improvement for the model.

3. **Model Aggregation :**

   The central server aggregates the computed gradients or model updates from the devices to create a new global model. Various aggregation methods, such as averaging or weighted averaging, can be used to combine the model updates. To preserve data privacy, federated learning employs techniques such as Differential Privacy (DP) or Secure Multi-Party Computation (SMC) during the aggregation process.

4. **Iterative Process :**

   The local model training and model aggregation steps are repeated iteratively, with each round improving the global model based on the collective knowledge of the participating devices. The central server communicates with the devices to synchronize the training process, exchange model updates, and provide instructions for the next round of training.

1.3 Benefits of Federated Learning :

Federated learning enables data owners to retain control over their data, making it suitable for sensitive data or compliance with data protection regulations. It leverages the computing power of multiple devices, allowing for efficient and scalable training on large data-sets without the need for data transfer. By training models on edge devices, federated learning facilitates real-time decision-making and reduces the reliance on cloud-based processing, making it suitable for applications with low latency requirements or limited network connectivity. Federated learning allows for training models on diverse data sources, leading to improved generalization and robustness.

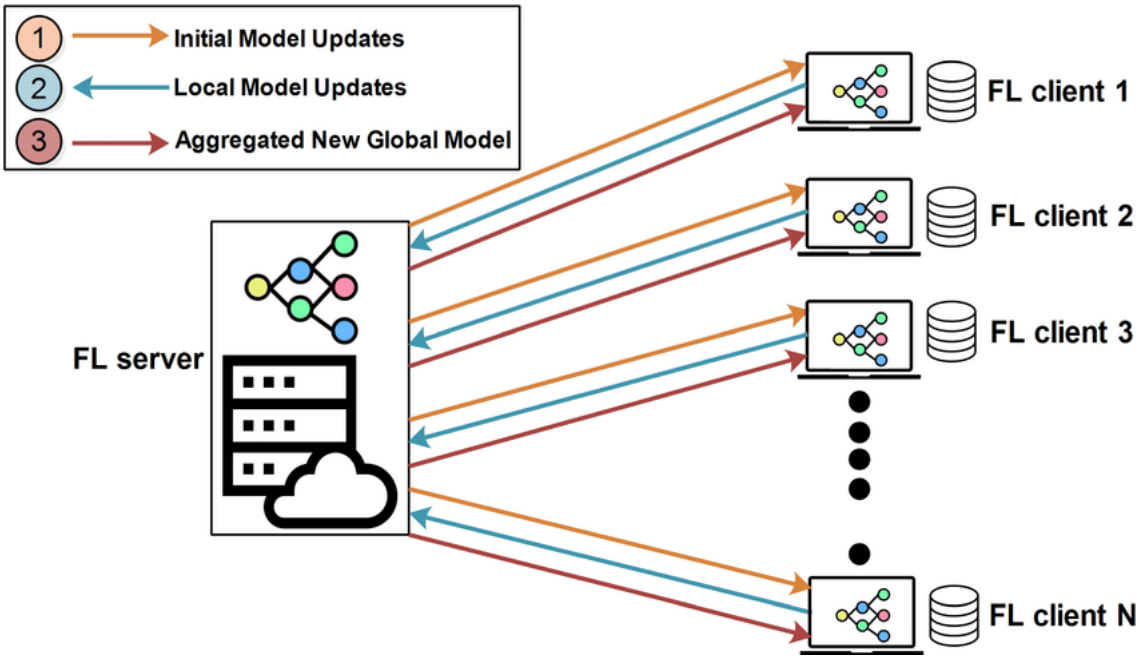

FIGURE 1.2 – General working process of federated learning [**fed**]

1.4 Characteristic of Federated Learning

Federated learning (FL) is distinct from traditional centralized machine learning approaches due to several key characteristics :

— FL involves decentralized data, where data remains on individual devices or edge nodes instead of being collected and stored in a central server. This prioritizes data privacy by avoiding the need for data sharing and keeping sensitive or personal data on devices.

— Collaborative learning is enabled across multiple devices or nodes, allowing for the aggregation of local model updates or gradients from diverse devices.

— It optimizes resource utilization by leveraging the computational capabilities of individual devices and employs model aggregation techniques to combine local model updates or gradients from participating devices.

— FL involves iterative rounds of local training and model aggregation, improving the global model over time.

— It optimizes resource utilization by leveraging the computational capabilities of individual devices and employs model aggregation techniques to combine local model updates or gradients from participating devices.

— Finally, FL accommodates scenarios with heterogeneous and unbalanced data distributions, making it suitable for applications with non-uniform or specialized data.

These characteristics make FL a powerful approach for collaborative machine learning, addressing challenges related to data privacy, scalability, and distributed learning. FL

enables the collective utilization of data resources, fosters privacy-aware collaborations, and supports intelligent decision-making at the edge.

1.5 Categorization of FL

Federated learning can be categorized into three main types :

1.5.1 Horizontal Federated Learning

Horizontal federated learning is a type of machine learning where data is distributed across multiple devices or nodes that have similar features. This approach is particularly useful when dealing with large datasets that cannot be processed on a single machine. By distributing the data across multiple devices, the computational load is shared, and the training process can be completed faster. One example of horizontal federated learning is a group of smartphones with similar hardware and software configurations participating in the training process. Each device in the group has access to a subset of the data, and the models are trained locally on each device. The local models are then aggregated to create a global model that is more accurate than any of the local models. Horizontal federated learning has several advantages over traditional machine learning approaches. First, it allows for the training of models on sensitive data without the need to transfer the data to a central location. This is particularly important in industries such as healthcare, where patient data must be kept confidential. Second, it reduces the risk of overfitting, as the models are trained on a diverse set of data. Finally, it allows for the training of models on data that is distributed across multiple locations, which can be useful in scenarios where the data cannot be centralized.

1.5.2 Vertical Federated Learning

In this approach, data is distributed across multiple devices or nodes that have different features, such as different patient populations in the case of hospitals. By pooling their data together, these organizations can develop more accurate and robust machine learning models that can be used to predict patient outcomes, identify disease patterns, and improve healthcare delivery. One of the key benefits of vertical federated learning is that it allows organizations to collaborate without sharing sensitive data. Instead of sending data to a central server, each organization keeps its data locally and only shares the necessary information with other nodes. This approach ensures that patient

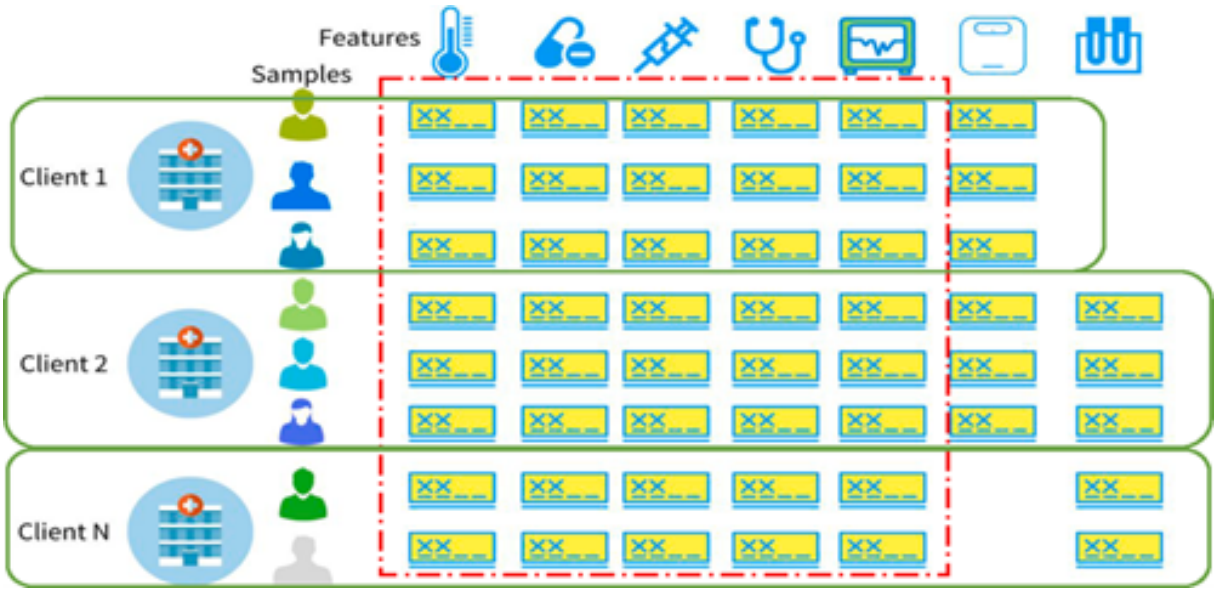

FIGURE 1.3 – An application sample of Horizontal FL.[9]

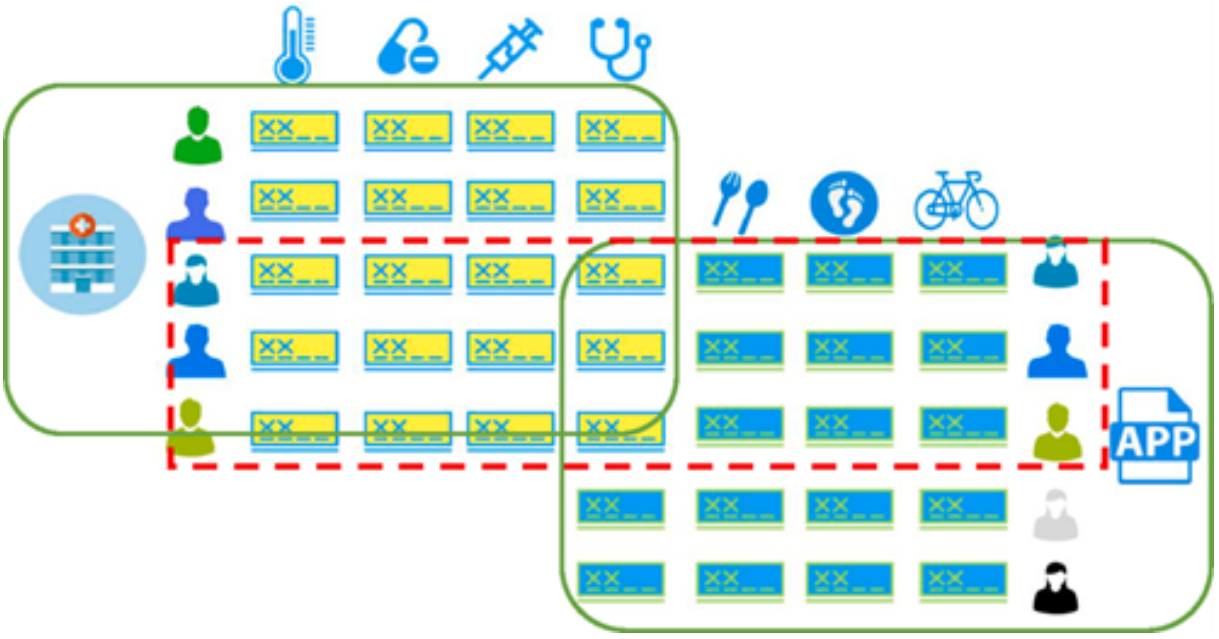

FIGURE 1.4 – An application sample of Vertical FL.[9]

privacy is protected, and that sensitive information is not exposed to unauthorized parties. Another advantage of vertical federated learning is that it allows organizations to leverage the strengths of different datasets. By combining data from multiple sources, organizations can develop more comprehensive models that are better able to capture the nuances of different patient populations. This can lead to more accurate predictions and better healthcare outcomes for patients.

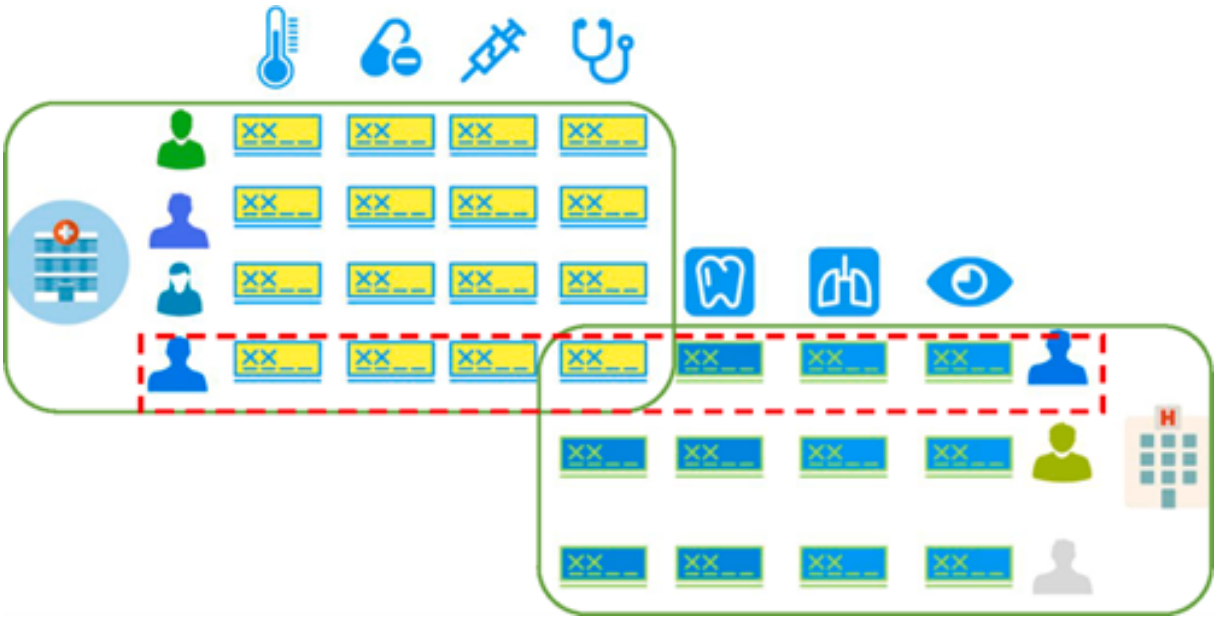

FIGURE 1.5 – An application sample of federated transfer learn.[9]

1.5.3 Federated Transfer Learning

Federated Transfer Learning is a highly effective technique for training machine learning models in a distributed setting. It harnesses the power of pre-trained models and fine-tuning on local data, significantly reducing the data requirements for training a model from scratch.

A major advantage of Federated Transfer Learning is its ability to produce more accurate models, even when individual device data is limited. The pre-trained model serves as a strong foundation, and fine-tuning allows customization to the specific data on each device. This approach improves model performance despite data constraints. Furthermore, Federated Transfer Learning addresses privacy concerns in distributed environments. By keeping data local to each device, sensitive information remains protected, while still enabling the creation of accurate models. This privacy-preserving aspect enhances the practicality and security of the technique.

1.6 Architectures of FL

1.6.1 Federated Averaging Architecture

The federated averaging architecture is the most widely used FL architecture. It consists of the following components :

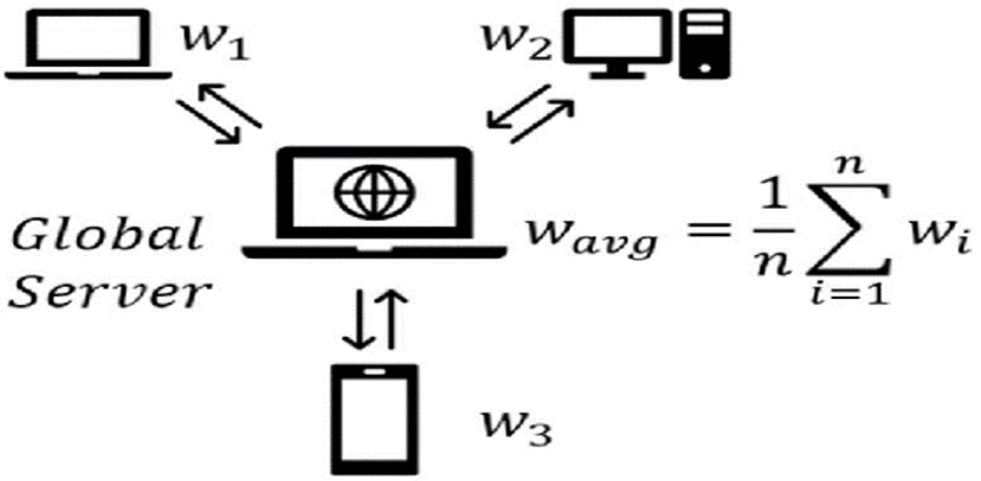

FIGURE 1.6 – Federated Averaging Architecture.[20]

— **Central Server :** The central server plays a crucial role in federated learning by overseeing the entire process. It starts by initializing the global model and then collects the model updates from all the participating devices. The server then aggregates these updates by averaging the local model updates or gradients to generate an updated global model.

— **Participating Devices :** Devices that are involved in the process, such as smartphones, IoT devices, or edge servers, store the data locally and conduct model training on their own. Every device trains the model using its own data, calculates model updates or gradients, and transmits them to the central server for consolidation.

— **Communication Protocol :**

A communication protocol is put in place to facilitate the exchange of model updates between the central server and the participating devices. The usual process involves the devices sending their updates to the central server, which in turn sends the updated global model for the next round of training.

1.6.2 Hierarchical Federated Learning Architecture

The FL process is structured hierarchically in the hierarchical federated learning architecture, which is designed to handle large-scale federated learning scenarios. This architecture comprises several levels of coordination and aggregation, and includes the following components :

— **Local Aggregators :**

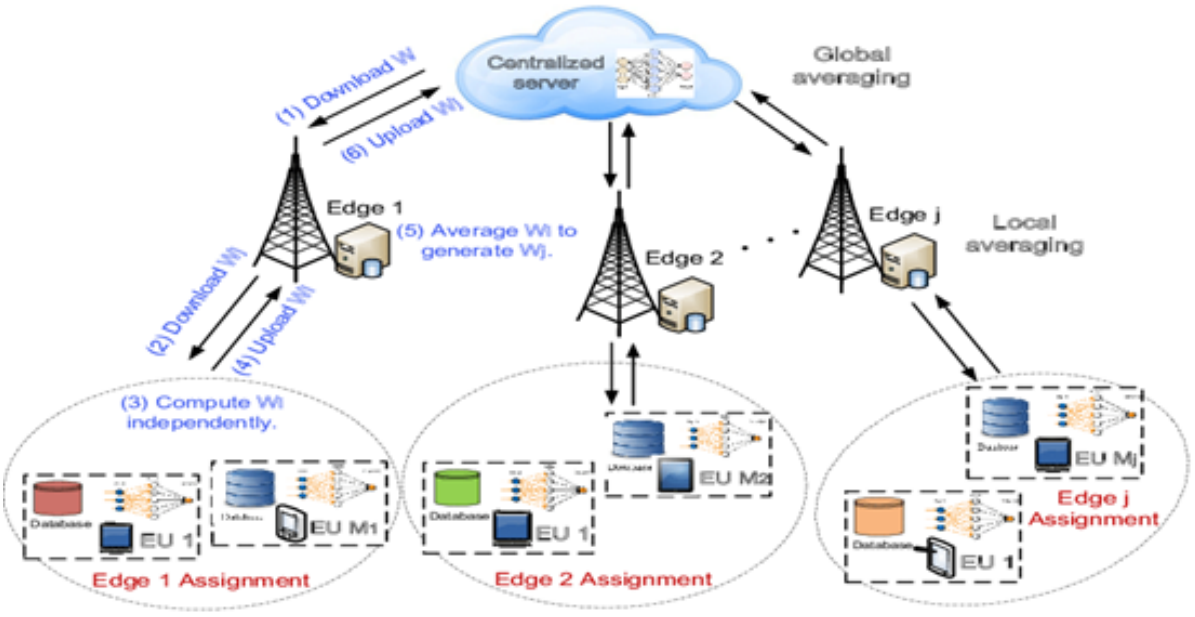

FIGURE 1.7 – Hierarchical federated learning : architecture and data flow [1]

local aggregators serving as intermediate nodes between participating devices and the central server. These local aggregators collect and aggregate model updates from a subset of devices within their respective local clusters.

— **Global Aggregator :**

The root aggregator, also referred to as the global aggregator, obtains the combined updates from the local aggregators. It conducts additional aggregation on the collected updates and produces an updated global model.

— **Communication Hierarchy :**

The communication within this architecture is organized in a hierarchical manner, where the local aggregators are responsible for communicating with the devices in their respective clusters and consolidating their updates. The global aggregator, on the other hand, communicates with the local aggregators to gather and combine the updates from all the local clusters.

1.6.3 Peer-to-Peer Federated Learning Architecture

The Peer-to-Peer (P2P) federated learning architecture utilizes a decentralized network, eliminating the need for a central server. Instead, participating devices communicate with each other for model updates and aggregation. This approach provides improved privacy and reduces dependence on a single point of coordination. The key components of this architecture include :

— **Participating Devices :**

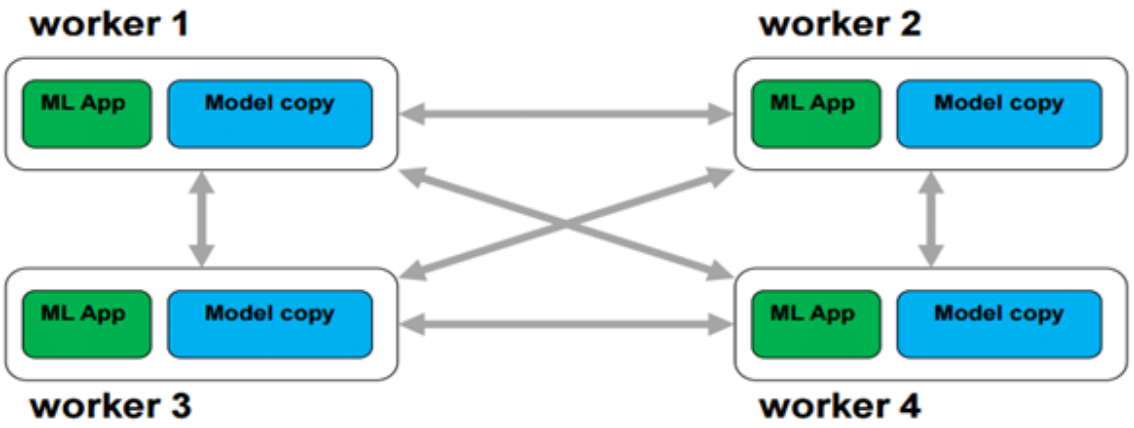

FIGURE 1.8 – Peer-to-peer (P2P) network topology for decentralized parameter storage. All workers may communicate with any other worker [2]

The devices in a peer-to-peer network establish direct communication with each other, maintaining their own local data and performing local model training. Through techniques such as averaging, weighted averaging, or secure multi-party computation, the devices exchange their model updates or gradients, which are then aggregated. (Niknam et al., 2020)

— **Decentralized Coordination :**

Gossip-based or overlay networks, which are peer-to-peer protocols, facilitate communication and coordination between devices. Each device is connected to a subset of other devices and disseminates updates across the network.

1.7 Open-source frameworks

Numerous open-source frameworks have been developed to assist researchers and developers in implementing federated learning into their projects. These frameworks are equipped with essential tools and infrastructure, each with strengths and weaknesses. Some of the most widely used frameworks, along with their key features and benefits, are highlighted below

1. **TensorFlow Federated (TFF) :**

   Developed by Google, TFF is an open-source framework for FL. It integrates with TensorFlow and provides high-level APIs for building FL algorithms. TFF supports federated aggregation, secure aggregation protocols, and advanced features like differential privacy.[17]

2. **PySyft :**

   PySyft is an open-source Python library built on top of PyTorch and supports FL using the "Federated Learning for Differential Privacy" (FLDP) concept. It

provides tools for secure and private computation, federated training, and federated evaluation.[13]

3. **Flower :**

   Flower is an open-source framework designed to simplify the development of FL systems. It provides a high-level API for building FL algorithms using TensorFlow or PyTorch. Flower supports various communication protocols, such as gRPC and WebSocket's, and offers fault tolerance and dynamic scalability.[3]

4. **FedML :**

   FedML is an open-source research library focusing on benchmarking and reproducibility of FL algorithms. It implements state-of-the-art FL algorithms across domains like FedAvg, FedProx, and FedNova. FedML supports TensorFlow and PyTorch and offers tools for evaluating FL models.[6]

5. **Federated AI Technology Enabler (FATE) :**

   FATE is an open-source project developed by Webank AI Department. It provides a secure computing framework for FL, including federated learning, transfer learning, and federated inference. FATE supports heterogeneous computing environments and offers privacy-preserving techniques, such as homomorphic encryption and secure multi-party computation. [5]

Frameworks such as those discussed earlier are essential for researchers and developers to use federated learning techniques effectively. These frameworks offer a variety of features and benefits that cater to different use cases and requirements. By using these open-source frameworks, researchers and developers can contribute to the progress of federated learning and create efficient machine-learning models that protect privacy.

1.8 Application of FL

1.8.1 Healthcare :

— Medical Research :

Federated learning facilitates collaborative research and analysis of medical data while safeguarding patient confidentiality. Numerous healthcare organizations can pool their data for joint model training without disclosing any sensitive patient data.

— Personalized Medicine :

FL enables the creation of customized treatment plans by utilizing data from various sources, including genomics, EHRs, wearables, and mobile health apps. This

allows for personalized healthcare interventions while maintaining the confidentiality of sensitive patient information.

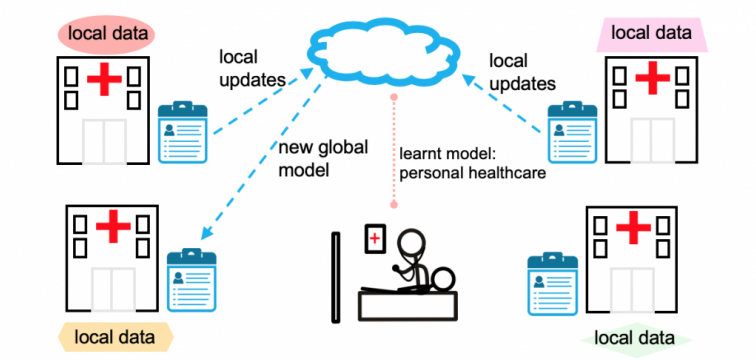

FIGURE 1.9 – application of federated learning for personal healthcare via learning over heterogeneous electronic medical records distributed across multiple hospitals.[18]

.

1.8.2 Internet of Things (IoT) :

FL enables IoT devices like sensors, wearables, and connected devices to engage in collaborative learning. These devices can learn from their local data and enhance their performance without the need to transmit sensitive data to a central server. In the context of smart cities, FL can be leveraged to create intelligent systems that analyse data from diverse sources such as traffic sensors, environmental sensors, and surveillance cameras. This collaborative analysis can optimize urban services and infrastructure.

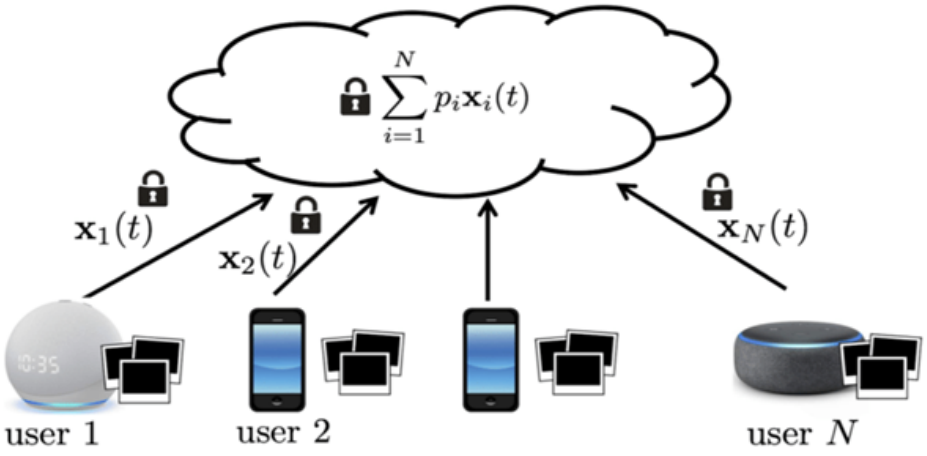

FIGURE 1.10 – Federated Learning for IoT Devices.[22]

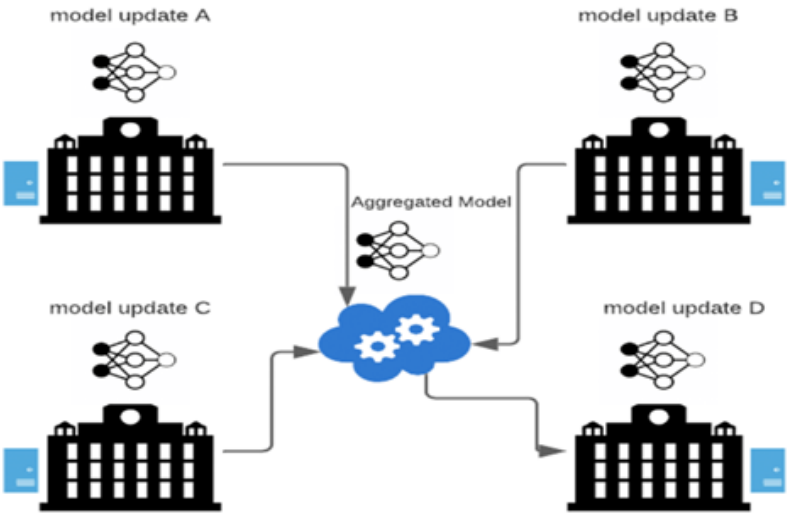

FIGURE 1.11 – Overview of Federated Learning across organisations[22]

1.8.3 Finance :

Federated learning is a powerful tool for detecting fraudulent activities in financial transactions by utilizing data from multiple banks or financial institutions. Through collaboration, patterns indicative of fraudulent behaviour can be identified while ensuring the protection of sensitive customer data. Additionally, FL enables financial institutions to collectively model risk factors and develop robust risk assessment models, improving the accuracy and effectiveness of credit scoring, insurance underwriting, and investment risk analysis.

1.8.4 Natural Language Processing (NLP) :

— Language Translation :

The utilization of FL in training language translation models allows for the incorporation of multilingual data from various sources, resulting in the creation of more precise and context-sensitive translation systems, all while ensuring data confidentiality.

— Speech Recognition :

Collaborative training of speech recognition models using data from various devices, including smartphones and smart speakers, is made possible by FL. This results in enhanced accuracy and performance of voice-enabled applications.

1.8.5 Autonomous Vehicles :

— Collaborative Learning :

Autonomous vehicles can leverage federated learning to exchange knowledge and insights while upholding privacy. By pooling their local sensor data, these vehicles can enhance their perception, decision-making, and control algorithms through collaborative learning.

— Traffic Prediction :

By analysing data from multiple vehicles or traffic management systems, FL can predict traffic patterns and congestion. This collaborative approach optimizes route planning and traffic flow without compromising individual vehicle data.

1.8.6 Energy and Environment :

— Energy Management :

In smart grids or energy systems, FL enables collaborative optimization of energy consumption and management. By leveraging local data, distributed devices can work together to enhance energy efficiency and demand response.

— Environmental Monitoring :

Collaborative analysis of environmental data, such as air quality or pollution levels, can be achieved using FL. By pooling data from multiple sensors or monitoring stations, accurate models for environmental monitoring and prediction can be developed.

1.9 Conclusion

In summary, federated learning (FL) is an advanced machine learning technique that enables collaborative model training while preserving data privacy and security. FL achieves this by decentralizing data on individual devices or edge nodes, allowing collective learning without transferring raw data to a central server. It offers benefits such as privacy preservation, optimized resource utilization, promotion of edge intelligence, and management of diverse and imbalanced data. FL has applications in healthcare, IoT, finance, NLP, autonomous vehicles, and energy/environment. It also provides different architectural configurations, including federated averaging, hierarchical FL, and peer-to-peer FL, allowing flexible implementation based on specific requirements. These architectures facilitate efficient communication and coordination between devices. In conclusion, federated learning strikes a balance between collective intelligence and data

privacy, and as it advances, we can expect further progress and improvements in the field, creating new opportunities for collaborative and privacy-conscious machine learning.

Chapitre 2

LITERATURE REVIEW

The literature review chapter of this essay critically examines a diverse range of scholarly sources that contribute to the understanding of the research topic, specifically the application of federated learning algorithms in a distributed learning environment. Numerous authors and researchers have explored this area, and their contributions are discussed in this chapter.

2.1 Motivation for Federated Learning

The motivation for adopting Federated Learning arises from the need to address challenges associated with data silos and data sensitivity in conventional machine learning approaches, as highlighted by T. Li, Sahu, Talwalkar, et al. (2020). Traditional methods involve collecting data from various sources and sending it to a central server for processing, which raises concerns about privacy, communication overhead, and data transfer requirements. Federated Learning tackles these issues by enabling participants to collaboratively build a shared model while ensuring the privacy of their local data. This approach allows sensitive data to remain on the devices, addressing privacy concerns and complying with data protection regulations. Additionally, it reduces communication overhead by transmitting only model updates instead of raw data, optimizing bandwidth usage and minimizing network latency. By enabling machine learning models to be trained on distributed data without centralizing it, Federated Learning offers a flexible and efficient solution for large-scale learning tasks. This research aims to leverage the insights provided by Li et al. to contribute to the practical implementation and further advancements of Federated Learning.

2.2 Trust issues in federated learning

Trust is a crucial concern in federated learning, and it is vital to address trust issues to ensure the reliability and security of this approach. Li et al. (2020) emphasize in their review of applications in federated learning that one of the key challenges is maintaining data privacy and preventing unauthorized access to sensitive information during the model training and aggregation process. Robust privacy-preserving mechanisms such as secure aggregation protocols (Kairoz et al., 2021), encryption techniques (Bonawitz et al., n.d.), and differential privacy (Abadi et al., 2016) are essential to safeguard data privacy and protect against potential vulnerabilities.

Ensuring model integrity and fairness is another critical trust-related issue in federated learning, as highlighted by Huang et al. (2020). Participants with diverse data sources may introduce biased or adversarial data, leading to inaccurate or unfair models. Rigorous model verification techniques and strategies are needed to address this concern.

Moreover, the central coordinating entity or server in federated learning needs to inspire confidence in its trustworthiness and security. To enhance trust, secure and verifiable aggregation protocols (Bonawitz et al., 2017), transparent governance mechanisms, and participant validation of the aggregation process are recommended (Kairouz et al., 2021).

Addressing trust issues in federated learning requires a multidimensional approach encompassing technical solutions, legal frameworks, and organizational policies. This includes developing secure algorithms, establishing trust frameworks and certifications, implementing transparent governance mechanisms, and fostering collaboration and trust-building among participants (Li et al., 2020). These collective efforts will contribute to making federated learning a reliable and trustworthy approach for collaborative machine learning in distributed environments.

2.3 Related work on trust mitigated federated learning

Federated learning has emerged as a promising approach for training machine learning models on distributed data without compromising data privacy. However, the trustworthiness of the participating clients in federated learning remains a critical concern. Malicious clients can intentionally or unintentionally introduce noise or bias into the model, leading to poor performance or even security breaches. To address this challenge, trust mitigated federated learning has been proposed as a solution that leverages trust mechanisms to ensure the reliability and integrity of the participating clients. In this

section, we will review the related work on trust mitigated federated learning and explore its potential to enhance the security and privacy of federated learning systems.

2.3.1 Securing Secure Aggregation : Mitigating Multi-Round Privacy Leakage in Federated Learning :

In order to address trust concerns in federated learning, researchers have proposed the use of secure aggregation techniques, as outlined in the study "Securing Secure Aggregation : Mitigating Multi-Round Privacy Leakage in Federated Learning by [16]. These protocols aim to safeguard the privacy and integrity of client updates during the aggregation process.Secure aggregation protocols ensure that the client updates remain encrypted and private throughout the aggregation process. To provide privacy guarantees, these protocols often incorporate the use of Differential Privacy (DP). By combining secure aggregation with DP, the privacy of individual client data can be protected.

2.3.2 Flexible Byzantine Fault Tolerance :

In their work titled "Flexible Byzantine Fault Tolerance [11], the authors address the challenge of designing a Byzantine fault-tolerant (BFT) consensus solution capable of withstanding higher corruption levels than those typically handled by the traditional Byzantine fault model. To tackle this issue, the authors introduce a novel approach called Flexible BFT, which is built on two fundamental principles : stronger resilience and diversity.

The first principle, stronger resilience, introduces a new fault model known as alive-but-corrupt faults. This model allows replicas to deviate from the protocol in an arbitrary manner, aiming to compromise safety. However, it ensures that replicas will not attempt to hinder the liveness of the protocol if they cannot compromise its safety. By incorporating this fault model, the protocol becomes more resilient and better equipped to withstand attacks.

The second principle, diversity, focuses on designing consensus solutions that employ the protocol transcript to generate different commit decisions based on diverse beliefs. This separation of beliefs allows the Flexible BFT solution to support both synchronous and asynchronous beliefs, as well as combinations of Byzantine and alive-but-corrupt fault resilience thresholds. This flexibility enables the protocol to adapt to various scenarios and requirements.

In conclusion, the authors propose a new approach to designing BFT consensus solutions that emphasizes stronger resilience and diversity. This approach enhances the protocol's

resilience and ability to withstand attacks while accommodating synchronous and asynchronous beliefs. The authors suggest that this approach has the potential to be applied beyond Byzantine fault tolerance, extending its benefits to other consensus protocols as well.

2.3.3 A Review on Various Applications of Reputation Based Trust Management :

The paper discusses the challenges of trust management in cloud computing, including issues with transparency, dynamism, and distribution. The authors propose a reputation-based trust management framework to handle trust in cloud environments. This framework includes a credibility model that differentiates between true and malicious feedback from consumers. The authors also developed a Trust Assessor tool to compare the trustworthiness of services and store the results in a database for future use. In conclusion, the paper highlights the importance of establishing trust between service providers and consumers in cloud computing. The proposed reputation-based trust management framework offers an efficient solution to this challenge by leveraging feedback from consumers to establish trustworthiness.[15]

2.3.4 PRIVACY-PRESERVING FEDERATED LEARNING BASED ON MULTI-KEY HOMOMORPHIC EN-CRYPTION :

The paper discusses the problems of data leakage and privacy breaches in federated learning scenarios, where multiple devices share the same encryption and decryption key. The authors propose a solution to these problems by applying multi-key homomorphic encryption, specifically the xMK-CKKS protocol. This protocol defines an aggregated public key and decryption share to achieve secure and simple encryption and decryption, which is more suitable for privacy protection in federated learning scenarios. In conclusion, the authors propose a novel privacy-preserving federated learning scheme based on multi-key homomorphic encryption to protect data privacy. They introduce xMK-CKKS as an improvement over MK-CKKS, which has the risk of privacy leakage when used in federated learning scenarios. The xMK-CKKS protocol provides stronger security than MK-CKKS and is also robust against any collusion between $k \leq N - 1$ honest-but-curious devices and the server. The authors suggest that this approach could have significant implications for mobile services and networks in the future.[10]

2.3.5 PPFL : Privacy-preserving Federated Learning with Trusted Execution Environments :

The paper discusses the problem of privacy leakage in federated learning and proposes a Privacy-preserving Federated Learning PPFL framework to address this issue. The paper highlight that traditional federated learning approaches can lead to privacy breaches as the model updates are sent in plaintext, which can be intercepted by adversaries. To solve this problem, the paper proposes a framework that utilizes Trusted Execution Environments (TEEs) on both clients and servers for local training and secure aggregation, respectively. This approach ensures that model/gradient updates are hidden from adversaries, thus limiting privacy leaks in federated learning. The proposed PPFL framework is based on greedy layer-wise training and aggregation, which overcomes the constraints posed by limited TEE memory while providing comparable accuracy of complete model training at the price of a tolerable delay. The layer-wise approach supports sophisticated settings such as training one or more layers (block) each time, potentially better dealing with heterogeneous data at the client-side and speeding up the training process. In conclusion, the authors present a practical framework that utilizes TEEs to limit privacy leaks in federated learning. Their implementation shows that this approach can significantly improve privacy while incurring small performance overhead. The proposed PPFL framework provides a promising solution for mobile systems where privacy is crucial.[12]

2.3.6 CRYPTEN : Secure Multi-Party Computation Meets Machine Learning :

The adoption of secure multi-party computation (MPC) in machine learning is limited due to the absence of flexible software frameworks that can integrate popular secure MPC primitives with modern machine learning frameworks. This limits the accessibility of secure MPC techniques to machine learning researchers and developers without a background in cryptography. The paper presents CRYPTEN, a flexible software framework that aims to make modern secure MPC techniques accessible to machine-learning researchers and developers without a background in cryptography. The framework exposes popular secure MPC primitives via abstractions that are common in modern machine-learning frameworks, such as tensor computations, automatic differentiation, and modular neural networks. The paper describes the design of CRYP TEN and measures its performance on state-of-the-art models for text classification, speech recognition, and image classification. The authors demonstrate that CRYP TEN's flexible PyTorch-like API makes private inference and training of modern machine-learning models easy to

implement and efficient. The paper concludes with a discussion of open problems and a roadmap for further development of CRYPTEN.[8]

2.4 Conclusion

In this chapter, we explored the literature related to Federated Learning, focusing on the motivation behind its adoption, trust issues, and related work on trust mitigated Federated Learning.

The reviewed literature highlights the importance of addressing trust issues from a multidimensional perspective, encompassing technical solutions, legal frameworks, and organizational policies. It emphasizes the need for secure algorithms, trust frameworks, transparent governance mechanisms, collaboration, and trust-building among participants.

Moving forward, further research and development are required to enhance the security, privacy, and trustworthiness of Federated Learning. The insights provided by the literature reviewed in this chapter serve as a foundation for the practical implementation and advancement of Federated Learning in real-world scenarios. By addressing trust concerns, Federated Learning can become a reliable and trustworthy approach for collaborative machine learning in distributed environments, enabling the development of robust and privacy-preserving models while respecting data privacy and protection regulations.

Chapitre 3

THREATS AND SECURITY IN FL

3.1 Privacy and Security Threats

Federated learning, as a distributed learning paradigm, introduces unique privacy and security challenges due to its collaborative nature involving multiple participants. This section focuses on exploring the key privacy and security threats associated with federated learning, which need to be addressed to ensure the confidentiality, integrity, and privacy of participants' data.

3.1.1 Data Leakage

One of the primary concerns in federated learning is the risk of data leakage. As the training process occurs locally on each participant's device or server, there is a potential for sensitive information to be exposed. Without appropriate privacy-preserving techniques, adversaries may attempt to infer private data samples from the global model's updates.

3.1.2 Membership Inference

Membership inference attacks aim to determine whether specific data points were present in a participant's training dataset. By analyzing the changes made to the global model during federated learning, an attacker may infer the presence or absence of certain data samples. This poses a significant privacy threat, particularly in sensitive domains where the mere knowledge of data membership can be detrimental.

3.1.3 Model Inversion

Model inversion attacks involve adversaries attempting to extract sensitive information from the trained model itself. By analyzing the model's outputs, gradients, or other information, attackers may gain insights into the participant's private training data. This threat becomes more prominent when the trained model is shared or deployed in untrusted environments.

3.1.4 Byzantine Attacks

Federated learning relies on the assumption that participants contribute honest and accurate model updates. However, adversaries may act maliciously and intentionally provide incorrect or manipulated updates to the global model. Such Byzantine attacks can compromise the quality and integrity of the trained model, leading to biased or misleading results.

3.2 Attack Models

Understanding different attack models is crucial for assessing the security risks in federated learning. Some common attack models include :

— Model Poisoning Attacks :

Attackers can manipulate the training data or the model update process to inject harmful samples or influence the learned parameters, which can compromise the accuracy of the model or introduce backdoors.

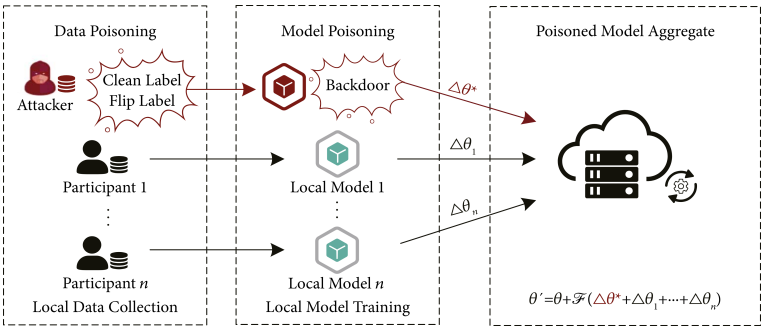

FIGURE 3.1 – Overview of the poisoning attacks against FL. The attacker pretends to be a benign participant, and shares crafted training data or deliberately tainted model updates to the aggregator.[21]

— Membership Inference Attacks :

By exploiting information leakage from the shared model's predictions, adversaries can determine if a particular sample was included in the training dataset, which violates the privacy of the participants.

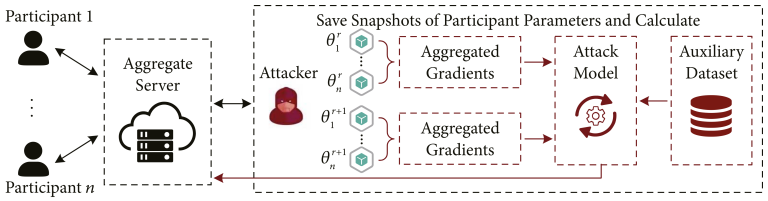

FIGURE 3.2 – Overview of inference attacks in FL. The attacker saves the snapshots of the aggregated model parameters in each round and performs inference attacks by employing the difference between the continuous snapshots.[21]

— Model Inversion Attacks :

Adversaries can reconstruct inputs or data samples that resulted model outputs, which may expose confidential information about the training data or individual participants.GAN-based model inversion attacks are a type of model inversion attack that uses generative adversarial networks (GANs) to generate synthetic data that is similar to the data used to train the model. GANs are a type of machine learning model that can be used to generate realistic images, text, and other types of data. By using a GAN, an attacker can generate synthetic data that is almost identical to the original data of the victim. This can be used to infer sensitive information from the victim.

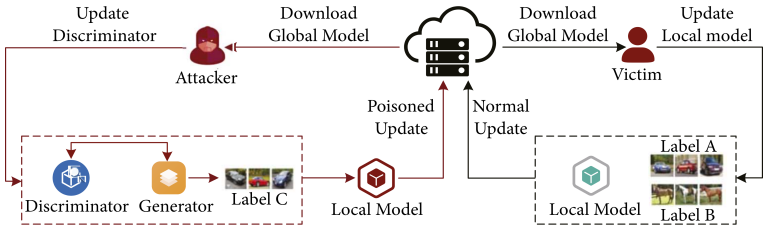

FIGURE 3.3 – By using the client-side GAN attacks, the attacker can reconstruct sensitive information from the victim.[21]

— Byzantine Attacks :

Adversaries may act in an arbitrary manner or collaborate with others to interfere with the federated learning process, thereby jeopardizing the shared model's accuracy and integrity.

3.3 Defense Mechanisms

To enhance the security of federated learning, various defense mechanisms can be employed :

3.3.1 Differential Privacy :

Differential privacy is a technique that enhances privacy by safeguarding the data of individual participants in federated learning. It employs a mathematical framework to measure and regulate the privacy guarantees provided by a learning algorithm or model. The core concept of differential privacy is to introduce controlled randomness or noise into the computation of the learning algorithm. This ensures that the output does not reveal any specific information about any individual participant's data, making it difficult for attackers to identify whether a particular individual's data was used in the training process or not. The privacy budget or privacy parameter is the foundation of differential privacy, which determines the level of privacy protection provided. A smaller privacy parameter provides stronger privacy guarantees but may also result in reduced accuracy or utility of the learning model. Therefore, finding the right balance between privacy and utility is a crucial consideration in the design of differentially private federated learning systems. There are various mechanisms and approaches to achieving differential privacy in federated learning, including Gaussian Noise Addition, Laplace Noise Addition, and Secure Multi-Party Computation (MPC). Gaussian noise and Laplace noise are commonly used methods for introducing privacy in federated learning, while MPC protocols allow participants to jointly compute the model updates or aggregate the results without revealing their individual contributions. Differential privacy provides a rigorous and provable privacy guarantee, ensuring that even with access to the output of the learning algorithm, it is challenging to infer sensitive information about individual participants. By incorporating differential privacy techniques into federated learning systems, organizations can demonstrate their commitment to privacy protection and build trust with data contributors. However, introducing noise for privacy preservation comes at the cost of utility or model accuracy. Striking the right balance between privacy and utility is a key challenge, and researchers continue to explore novel techniques and optimizations to improve the trade-off between privacy and model performance in federated learning. [19]

3.3.2 Secure Aggregation :

Secure aggregation protocols play a critical role in federated learning by ensuring that model updates from participants are combined securely without revealing individual contributions. These protocols aim to protect the privacy and integrity of participants' data during the aggregation process, mitigating the risk of attacks and information leakage. The aggregation process in federated learning involves collecting the model updates from multiple participants and merging them to create a global model. However, traditional aggregation methods may expose sensitive information, as adversaries could potentially infer participants' data by analyzing the aggregated updates. Secure aggregation protocols employ cryptographic techniques to enable privacy-preserving and tamper-resistant aggregation. They provide strong security guarantees by ensuring that the aggregated model reflects the contributions of all participants while preserving the privacy of their individual updates. Here are some common techniques used in secure aggregation protocols :

3.3.2.1 Homomorphic Encryption :

Homomorphic encryption allows computations to be performed directly on encrypted data without decrypting it. Secure aggregation protocols leverage homomorphic encryption to encrypt the participants' model updates before sending them to the aggregator. The aggregator can perform computations on the encrypted updates and generate an encrypted aggregated model. Finally, the aggregated model is decrypted to obtain the final global model. The use of homomorphic encryption ensures that the participants' updates remain private throughout the aggregation process.

3.3.2.2 Secret Sharing :

Secret sharing is a technique where a secret is divided into multiple shares distributed among participants. Secure aggregation protocols use secret sharing to divide the model updates into shares and distribute them among the participants. During the aggregation, the participants collaboratively combine their shares to compute the aggregated model without any participant revealing their individual update. Secret sharing provides protection against attacks, as an attacker would need to compromise multiple participants to obtain the complete model update.[4]

3.3.2.3 Secure Multi-Party Computation MPC :

MPC protocols enable participants to jointly compute functions or operations on their private inputs without revealing those inputs to each other. In the context of secure aggregation, MPC protocols are used to securely aggregate the participants' model updates. Each participant locally processes their update in a way that conceals their input, and the aggregated result is obtained without exposing individual contributions. MPC ensures that the aggregation process remains secure and private, even when participants are potentially untrusted or compromised.

By employing secure aggregation protocols, federated learning systems can effectively protect against attacks during the aggregation process. These protocols ensure that participants' data remains confidential and secure, preventing adversaries from extracting sensitive information or manipulating the aggregated model to undermine the learning process. However, it is important to carefully design and implement secure aggregation protocols to ensure their effectiveness. Adversaries may attempt various attacks, such as Byzantine attacks, where malicious participants intentionally provide incorrect or manipulated updates. Therefore, robust security measures, such as redundancy checks, error detection, and anomaly detection, should be incorporated into the protocols to detect and mitigate such attacks. Overall, secure aggregation protocols provide a crucial security mechanism in federated learning, enabling participants to collaborate while preserving the privacy and integrity of their data throughout the aggregation process.

Chapitre 4

Implementation and Results

4.1 System design and architecture

4.1.1 Data Partitioning and Distribution :

— Data partitioning is a crucial aspect of federated learning. It involves dividing the data among multiple clients while preserving data ownership and privacy.

— Partitioning strategies can be based on various factors, such as client devices, geographic locations, or specific data characteristics.

— Partitioning ensures that sensitive user data remains on the clients, reducing the risk of data exposure and complying with privacy regulations

4.1.2 Client-Side Execution :

— In federated learning, clients perform local training on their respective datasets using their computational resources, such as CPUs or GPUs.

— TensorFlow Federated enables the execution of federated computations on client devices, allowing clients to train models using their local data

— - Client-side execution ensures that sensitive data remains on the clients' devices, preventing the need for centralized data storage and reducing privacy risks.

4.1.3 Server-Side Aggregation :

— After local training, clients send their model updates or gradients to the server for aggregation.

— The server aggregates the model updates using secure aggregation protocols while preserving privacy and maintaining the confidentiality of individual client contributions.

— Secure aggregation techniques, such as cryptographic protocols (e.g., secure multi-party computation) or trusted execution environments (TEEs), can be used to protect the privacy and integrity of the model updates during aggregation.

4.1.4 Logging and Visualization :

— Logging mechanisms capture relevant metrics, such as accuracy and loss, during the federated learning process

— These metrics are logged at regular intervals and stored in a centralized log or database for analysis and monitoring.

— Visualization techniques, using libraries like Matplotlib and Seaborn, are utilized to create graphical representations of the training progress, model performance, and privacy measures.

— Stakeholders can leverage these visualizations to gain insights into the learning process, track performance trends, and make informed decisions.

4.1.5 Scalability and Robustness :

— The system design should be scalable to handle a large number of clients and varying computational resources and network conditions.

— Load balancing mechanisms can be implemented to distribute the computational load across multiple servers and ensure efficient training.

— The system should be robust, capable of handling client failures, network disruptions, or unexpected events without compromising privacy or interrupting the learning process.

— Techniques like client selection strategies, fault tolerance mechanisms, and adaptive training processes can enhance system robustness and performance.

In summary, the system design and architecture ensure the privacy-preserving nature of federated learning through techniques such as client-side execution, secure communication, and differential privacy integration. The logging and visualization components offer transparency and insights into the training process, facilitating analysis and decision-making. Scalability and robustness considerations enable the system to handle many clients and varying conditions while maintaining data privacy and integrity.

4.2 Data privacy and security measures

Data privacy and security measures are strategies and practices implemented to protect sensitive data from unauthorized access, disclosure, alteration, or destruction. In the context of this essay, the following data privacy and security measures are relevant :

4.2.1 `dp_accounting`

`dp_accounting` is a library that provides tools for calibrating and accounting for the privacy parameters in differentially private algorithms. It offers functionalities to compute privacy budgets, determine noise levels, and manage privacy guarantees based on the desired level of privacy.

Differential Privacy Algorithm implmented by `dp_accounting` :

Algorithm 1 Differential Privacy Algorithm

Require: ε : privacy parameter
Require: δ : privacy parameter
Require: $\mathcal{D}$: dataset
1: Initialize an empty array $\mathcal{M}$ to store the modified dataset
2: **for** each record x in $\mathcal{D}$ **do**
3: Generate a random noise N from a Laplace distribution : $N \sim \text{Lap}(\frac{\Delta f}{\varepsilon})$, where Δf is the sensitivity of the function f
4: Add the noise to the record : $x' \leftarrow x + N$
5: Add x' to $\mathcal{M}$
6: **end for**
7: **return** $\mathcal{M}$

4.2.2 `tff.learning.model_update_aggregator.dp_aggregator`

`tff.learning.model_update_aggregator.dp_aggregator` is a function provided by TensorFlow Federated (TFF) that enables differential privacy with adaptive clipping during the model aggregation process. It applies differential privacy mechanisms to the aggregated model updates, preventing the disclosure of sensitive information from individual clients.

4.2.3 Preprocessing Functions :

```
def get_emnist_dataset():
  emnist_train, emnist_test = tff.simulation.datasets.emnist.load_data(
      only_digits=True)

  def element_fn(element):
    return collections.OrderedDict(
        x=tf.expand_dims(element['pixels'], -1), y=element['label'])

  def preprocess_train_dataset(dataset):
    return (dataset.map(element_fn)
                   .shuffle(buffer_size=418)
                   .repeat(1)
                   .batch(32, drop_remainder=False))

  def preprocess_test_dataset(dataset):
    return dataset.map(element_fn).batch(128, drop_remainder=False)

  emnist_train = emnist_train.preprocess(preprocess_train_dataset)
  emnist_test = preprocess_test_dataset(
      emnist_test.create_tf_dataset_from_all_clients())
  return emnist_train, emnist_test

train_data, test_data = get_emnist_dataset()
```

The `preprocess_train_dataset` and `preprocess_test_dataset` functions are responsible for preparing the training and testing datasets, respectively, before they are used in the federated learning process. These functions typically apply data transformations, such as reshaping, shuffling, and batching, to ensure the data is in the appropriate format for training while preserving privacy.

4.3 Results and Analysis

1. **Accuracy Trend :**

   As the training rounds progress, there is a general upward trend in accuracy for all noise multiplier values. At the beginning of the training process, the accuracy is relatively low for all noise multiplier values. This is expected as the model is initially untrained. As the training continues, the accuracy improves steadily for

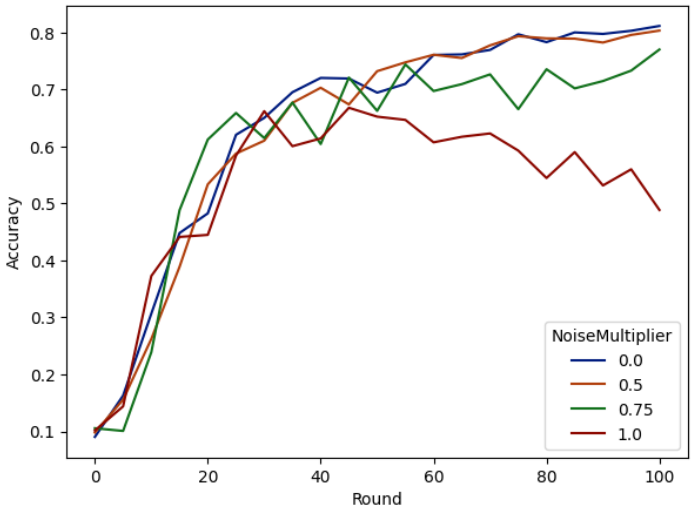

FIGURE 4.1 – accuracy plotting result

each noise multiplier value. Higher noise multiplier values (e.g., 0.75 and 1.0) show a slightly slower improvement in accuracy compared to lower noise multipliers (e.g., 0.0 and 0.5). This suggests that higher noise levels may introduce more challenges for the model to learn effectively. Towards the later rounds of training, the accuracy tends to plateau or reach a stable level for all noise multiplier values.

2. **Loss Trend :**

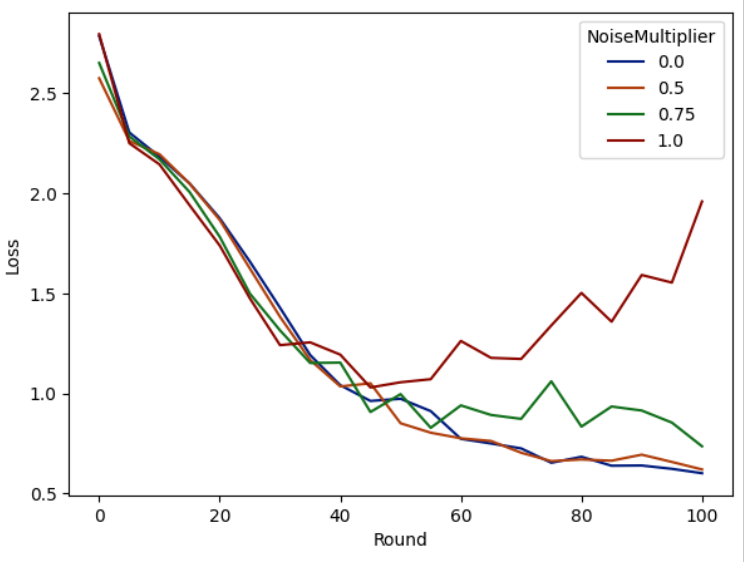

FIGURE 4.2 – loss plotting result

The loss values show a decreasing trend as the training progresses for all noise 44 multiplier values. Initially, the loss is relatively high for all noise multipliers, indicating that the model makes significant errors during the early stages of training. As the training continues, the loss gradually decreases, indicating that the model

learns to make better predictions and reduces its overall error. Higher noise multiplier values exhibit slightly higher loss values compared to lower noise multipliers throughout the training process, suggesting that higher noise levels may hinder the model's ability to minimize its loss effectively.

Overall, the trends observed in the graph indicate that the model's performance improves with more training rounds, regardless of the noise multiplier. However, higher noise levels can introduce additional challenges, leading to slower improvements in accuracy and relatively higher loss values. It's important to strike a balance between noise levels and model performance to achieve the desired level of accuracy while maintaining acceptable loss values.

the previous process estimate that for reaching the desired (2, 1e-05)-DP settings we need to use, use 120 clients with noise multiplier 1.2 so the final model parameters as blow :

```
[ ]  rounds = 100
     noise_multiplier = 1.2
     clients_per_round = 120

     data_frame = pd.DataFrame()
     data_frame = train(rounds, noise_multiplier, clients_per_round, data_frame)

     make_plot(data_frame)
```

FIGURE 4.3 – final private model parameters

As we can see, the final model has similar loss and accuracy to the model trained without noise, but this one satisfies (2, 1e-5)-DP :

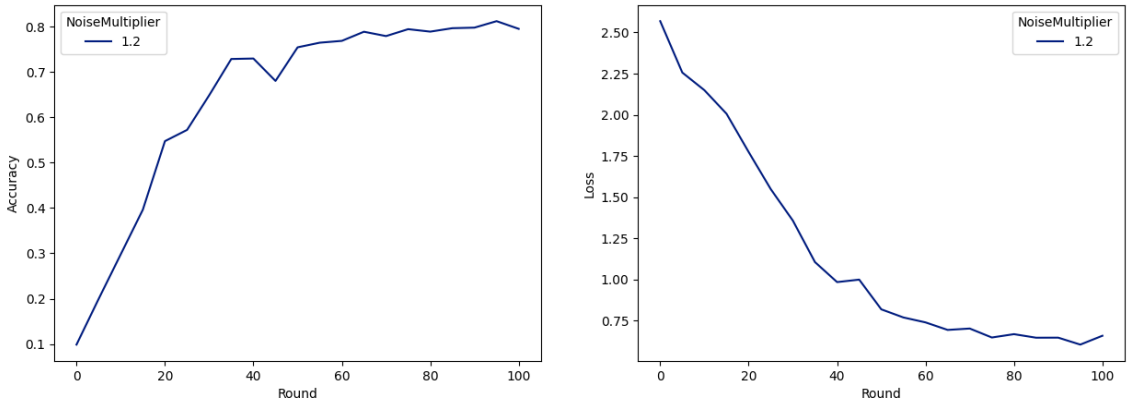

FIGURE 4.4 – final model result

Conclusion

In this essay, the analysis of trust-mitigated federated learning has provided significant insights to the field. The findings demonstrate the effectiveness of trust mitigation techniques, specifically noise multipliers, in improving model accuracy and convergence while preserving data privacy. The performance evaluation offers valuable metrics such as accuracy, loss, and convergence rate, enabling researchers and practitioners to assess the approach's efficiency and effectiveness. By comparing trust-mitigated federated learning with traditional methods, the study highlights the trade-offs between privacy preservation and model performance, providing benchmarking and understanding of trust mitigation techniques. The practical implications are relevant to industries and organizations seeking privacy-preserving solutions, showcasing the benefits of federated learning in privacy preservation, enhanced trustworthiness, and balancing privacy and performance requirements. The research also identifies future directions, including exploring advanced trust mitigation techniques, addressing scalability challenges, and adapting the approach to diverse data types and domains. In conclusion, this essay contributes to advancing federated learning by providing insights into its performance, practical implications, and potential for enhancing privacy and trust in machine learning systems. These findings can guide the development of secure and privacy-preserving solutions, promoting responsible data collaborations across various domains.

Bibliographie

[1] Alaa Awad Abdellatif et al. *Communication-Efficient Hierarchical Federated Learning for IoT Heterogeneous Systems with Imbalanced Data*. en. arXiv :2107.06548 [cs]. Juill. 2021. URL : http://arxiv.org/abs/2107.06548 (visité le 14/06/2023).

[2] Ons Aouedi, Kandaraj Piamrat et Benoît Parrein. "Intelligent Traffic Management in Next-Generation Networks". In : *Future Internet* 14 (jan. 2022), p. 44. DOI : 10.3390/fi14020044.

[3] Daniel J. Beutel et al. *Flower : A Friendly Federated Learning Research Framework*. arXiv :2007.14390 [cs, stat]. Mars 2022. URL : http://arxiv.org/abs/2007.14390 (visité le 12/06/2023).

[4] Keith Bonawitz et al. *Practical Secure Aggregation for Privacy Preserving Machine Learning*. URL : https://eprint.iacr.org/undefined/undefined (visité le 14/06/2023).

[5] *Fate*. URL : https://fate.fedai.org/ (visité le 12/06/2023).

[6] *FedML - Open and Collaborative Machine Learning Platform and AI x Web3 Marketplace*. URL : https://www.fedml.ai/ (visité le 12/06/2023).

[7] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Second edition. Beijing [China] ; Sebastopol, CA : O'Reilly Media, Inc, 2019. ISBN : 9781492032649.

[8] Brian Knott et al. *CrypTen : Secure Multi-Party Computation Meets Machine Learning*. arXiv :2109.00984 [cs]. Sept. 2022. URL : http://arxiv.org/abs/2109.00984 (visité le 12/06/2023).

[9] Li Li et al. "A review of applications in federated learning". en. In : *Computers & Industrial Engineering* 149 (nov. 2020), p. 106854. ISSN : 03608352. DOI : 10.1016/j.cie.2020.106854. URL : https://linkinghub.elsevier.com/retrieve/pii/S0360835220305532 (visité le 12/06/2023).

[10] Jing Ma et al. *Privacy-preserving Federated Learning based on Multi-key Homomorphic Encryption*. arXiv :2104.06824 [cs] version : 1. Avr. 2021. URL : http://arxiv.org/abs/2104.06824 (visité le 12/06/2023).

[11] Dahlia MALKHI, Kartik NAYAK et Ling REN. *Flexible Byzantine Fault Tolerance*. arXiv :1904.10067 [cs] version : 2. Mai 2019. URL : http://arxiv.org/abs/1904.10067 (visité le 12/06/2023).

[12] Fan MO et al. *PPFL : Privacy-preserving Federated Learning with Trusted Execution Environments*. arXiv :2104.14380 [cs] version : 2. Juin 2021. URL : http://arxiv.org/abs/2104.14380 (visité le 12/06/2023).

[13] *OpenMined*. URL : https://www.openmined.org/ (visité le 14/06/2023).

[14] Amir Masoud RAHMANI et al. "Machine Learning (ML) in Medicine : Review, Applications, and Challenges". en. In : *Mathematics* 9.22 (nov. 2021), p. 2970. ISSN : 2227-7390. DOI : 10.3390/math9222970. URL : https://www.mdpi.com/2227-7390/9/22/2970 (visité le 24/06/2023).

[15] Govindaraj RAMYA et al. "A Review on Various Applications of Reputation Based Trust Management". en. In : *International Journal of Interactive Mobile Technologies (iJIM)* 15.10 (mai 2021), p. 87. ISSN : 1865-7923. DOI : 10.3991/ijim.v15i10.21645. URL : https://online-journals.org/index.php/i-jim/article/view/21645 (visité le 12/06/2023).

[16] Jinhyun SO et al. *Securing Secure Aggregation : Mitigating Multi-Round Privacy Leakage in Federated Learning*. arXiv :2106.03328 [cs, math]. Juin 2021. URL : http://arxiv.org/abs/2106.03328 (visité le 12/06/2023).

[17] *TensorFlow*. URL : https://www.tensorflow.org/ (visité le 14/06/2023).

[18] Machine Learning Department UNIVERSITY Carnegie Mellon. *Federated Learning : Challenges, Methods, and Future Directions*. en-US. Section : Educational. Nov. 2019. URL : https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/ (visité le 14/06/2023).

[19] Kang WEI et al. "Federated Learning With Differential Privacy : Algorithms and Performance Analysis". en. In : *IEEE Transactions on Information Forensics and Security* 15 (2020), p. 3454-3469. ISSN : 1556-6013, 1556-6021. DOI : 10.1109/TIFS.2020.2988575. URL : https://ieeexplore.ieee.org/document/9069945/ (visité le 12/06/2023).

[20] Jihwan WON. *Abnormal Local Clustering in Federated Learning*. Août 2022.

[21] Junpeng ZHANG et al. "Security and Privacy Threats to Federated Learning : Issues, Methods, and Challenges". en. In : *Security and Communication Networks* 2022 (sept. 2022). Sous la dir. de Zhen WANG, p. 1-24. ISSN : 1939-0122, 1939-0114. DOI : 10.1155/2022/2886795. URL : https://www.hindawi.com/journals/scn/2022/2886795/ (visité le 14/06/2023).

[22] Tuo ZHANG et al. *Federated Learning for Internet of Things : Applications, Challenges, and Opportunities*. en. arXiv :2111.07494 [cs]. Avr. 2022. URL : http://arxiv.org/abs/2111.07494 (visité le 12/06/2023).