

Springer Optimization and Its Applications 213

My T. Thai

Hai N. Phan

Bhavani Thuraisingham *Editors*

Handbook of Trustworthy Federated Learning


 Springer

Springer Optimization and Its Applications

Volume 213

Series Editors

Panos M. Pardalos , University of Florida, Gainesville, USA

My T. Thai , CSE Building, University of Florida, Gainesville, USA

Advisory Editors

Roman V. Belavkin, Faculty of Science and Technology, Middlesex University, London, UK

R.D. Deshpande, University of Chicago, Chicago, USA

Vipin Kumar, Dept Comp Sci & Engg, University of Minnesota, Minneapolis, USA

Anna Nagurney, Isenberg School of Management, University of Massachusetts Amherst, Amherst, USA

Jun Pei, School of Management, Hefei University of Technology, Hefei, China

Oleg Prokopyev, Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, USA

Mauricio Resende, Amazon (United States), Seattle, USA

Van Vu, Department of Mathematics, Yale University, New Haven, USA

Michael N. Vrahatis, Mathematics Department, University of Patras, Patras, Greece

Guoliang Xue, Ira A. Fulton School of Engineering, Arizona State University, Tempe, USA

Yinyu Ye, Stanford University, Stanford, USA

Honorary Editor

Ding-Zhu Du, University of Texas at Dallas, Richardson, TX, USA

Aims and Scope

Optimization has continued to expand in all directions at an astonishing rate. New algorithmic and theoretical techniques are continually developing and the diffusion into other disciplines is proceeding at a rapid pace, with a spot light on machine learning, artificial intelligence, and quantum computing. Our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in areas not limited to applied mathematics, engineering, medicine, economics, computer science, operations research, and other sciences.

The series **Springer Optimization and Its Applications (SOIA)** aims to publish state-of-the-art expository works (monographs, contributed volumes, textbooks, handbooks) that focus on theory, methods, and applications of optimization. Topics covered include, but are not limited to, nonlinear optimization, combinatorial optimization, continuous optimization, stochastic optimization, Bayesian optimization, optimal control, discrete optimization, multi-objective optimization, and more. New to the series portfolio include Works at the intersection of optimization and machine learning, artificial intelligence, and quantum computing.

Volumes from this series are indexed by Web of Science, zbMATH, Mathematical Reviews, and SCOPUS.

My T. Thai • Hai N. Phan • Bhavani Thuraisingham
Editors

Handbook of Trustworthy Federated Learning

 Springer

Editors

My T. Thai 
Department of Computer & Information
Science & Engineering
University of Florida
Gainesville, FL, USA

Hai N. Phan
Department of Data Science
New Jersey Institute of Technology
Newark, NY, USA

Bhavani Thuraisingham
Department of Computer Science
The University of Texas at Dallas
Richardson, TX, USA

ISSN 1931-6828 ISSN 1931-6836 (electronic)
Springer Optimization and Its Applications
ISBN 978-3-031-58922-5 ISBN 978-3-031-58923-2 (eBook)
<https://doi.org/10.1007/978-3-031-58923-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

Federated Learning, standing at the intersection of collaborative knowledge learning and preserving individual data privacy, has emerged as a transformative paradigm in the dynamic landscape of machine learning. The origins of Federated Learning can be traced back to the early 2010s, a period coinciding with the proliferation of mobile devices. Traditional machine learning models faced a significant challenge in adapting to the decentralized nature of data generated by these devices. The pursuit of collaborative learning while preserving user data privacy gained momentum during this era, leading to the conceptualization of federated approaches. In 2016, Google researchers made a groundbreaking contribution by introducing the term “Federated Learning,” positioning it as a solution for training machine learning models across decentralized edge devices. This groundbreaking concept allowed devices to collaboratively learn a shared model while keeping raw data localized, thus promising to protect data privacy.

Since those pioneering days, Federated Learning has undergone a remarkable evolution. The field has matured with an influx of research, addressing challenges ranging from privacy concerns and security vulnerabilities to optimizing model performance in distributed settings. Importantly, Trustworthy Federated Learning, as a concept, evolved organically in response to the growing recognition that Federated Learning’s promise of collaborative learning was inseparable from the imperatives of privacy preservation and model security. The increasing demand for Trustworthy Federated Learning coincided with an era where data privacy and protection gained unprecedented prominence.

This handbook encapsulates this evolution, offering insights into the diverse facets of FL and its pivotal role in shaping the future of collaborative and trustworthy machine learning. It is an effort to be a reliable resource for researchers, academics, and practitioners deeply engaged or venturing into the realms of Trustworthy Federated Learning. Each chapter, in survey or expository form, is self-contained, making it easy for reading. There are 14 chapters, organized into 4 parts:

- **Part I: Security and Privacy** scrutinizes the intricate interplay between trustworthiness, privacy, and security within Federated Learning. These chapters not only explore robust defense mechanisms against targeted attacks but also address fairness concerns, providing a multifaceted foundation for securing Federated Learning systems against evolving threats. The historical context of privacy regulations, such as the European Union’s General Data Protection Regulation (GDPR), underscores the relevance and urgency of these discussions in contemporary settings.
- **Part II: Bilevel Optimization** immerses readers in the nuanced world of federated bilevel optimization. With a focus on applications such as hyperparameter optimization and neural network architecture search, these chapters unravel the intricacies of optimizing performance in federated settings. The historical perspective reveals the evolution of optimization challenges, mirroring the trajectory of Federated Learning from its nascent stages to its current applications.
- **Part III: Graph and Large Language Models** walks readers to the cutting edge of Federated Graph Learning. Addressing challenges in training Graph Neural Networks and ensuring privacy in Federated Learning of natural language models, these chapters spotlight the transformative potential of FL in graph-related tasks and large language models. The historical journey continues, highlighting the pivotal role of FL in advancing graph-related machine learning applications and also to the recent success of pre-trained large models.
- **Part IV: Edge Intelligence and Applications** introduces pioneering concepts such as Edge Federated Learning and Zone-Based Federated Learning. These chapters demonstrate how Federated Learning can empower mobile applications and preserve privacy with synthetic data. The historical exploration culminates in discussions about the evolution of edge computing and its integration with Federated Learning paradigms. The book concludes with Chap. 14, where the emerging landscape of Green Federated Learning takes center stage. It explores the intricate trade-offs between computing, learning algorithms, and communication systems, framing Green Federated Learning as a response to contemporary challenges in energy-intensive model training.

As the field continues to evolve, we do not intend to cover every possible issue in Trustworthy Federated Learning, but instead, present areas that we think are the most beneficial to the readers in getting the first overall picture of the field. It is our hope that the insights, strategies, and innovations presented within these pages not only deepen understanding but also propel the field of Trustworthy Federated Learning into new realms of trust, responsibility, and transformative impact.

We would like to extend gratitude to the 47 authors/co-authors who poured their expertise and passion into shaping this handbook. Their commitment to advancing the field of Federated Learning is evident in the richness and depth of the content presented within these pages. It would not have been possible without their high-quality work. We would like to thank the Springer editorial team, especially Elizabeth Loew, for their help and support in publishing this book. This project is

partially supported by the National Science Foundation under grants SaCT 1935928, SaCT 1935923, SaCT 2140477, and DGE 1723602.

Gainesville, FL, USA
Newark, NJ, USA
Richardson, TX, USA
December 2023

My T. Thai
Hai N. Phan
Bhavani Thuraisingham

Contents

Part I Security and Privacy

Trustworthiness, Privacy, and Security in Federated Learning	3
Sisi Zhou, Lijun Xiao, Yufeng Xiao, and Meikang Qiu	
Secure Federated Learning	39
Bo Tang and Xingyu Li	
Data Poisoning and Leakage Analysis in Federated Learning	73
Wenqi Wei, Tiansheng Huang, Zachary Yahn, Anoop Singhal, Margaret Loper, and Ling Liu	
Robust Federated Learning Against Targeted Attackers Using Model Updates Correlation	109
Priyesh Ranjan, Ashish Gupta, and Sajal K. Das	
Unfair Trojan: Targeted Backdoor Attacks Against Model Fairness	149
Nicholas Furth, Abdallah Khreishah, Guanxiong Liu, NhatHai Phan, and Yasser Jararweh	

Part II Bi-level Optimization

Federated Bilevel Optimization	171
Hongchang Gao	
A Two-Stage Stochastic Programming Approach for the Key Management q-Composite Scheme	197
Maciej Rysz, Guanglin Xu, and Alexander Semenov	

Part III Graph and Large Language Models

Recent Advances in Federated Graph Learning	223
Tre' R. Jeter and My T. Thai	

Privacy in Federated Learning Natural Language Models	259
Phung Lai and C. Ariel Pinto	
Federated Learning of Models Pretrained on Different Features with Consensus Graphs	289
Tengfei Ma, Jie Chen, and Trong Nghia Hoang	
Part IV Edge Intelligence and Applications	
Robust Federated Learning for Edge Intelligence	323
Dongxiao Yu, Xiao Zhang, Hanshu He, Shuzhen Chen, Jing Qiao, Yangyang Wang, and Xiuzhen Cheng	
ZoneFL: Zone-Based Federated Learning at the Edge	367
Xiaopeng Jiang, Hessamaldin Mohammadi, Cristian Borcea, and NhatHai Phan	
Synthetic Data for Privacy Preservation in Distributed Data Analysis Systems	393
Anantaa Kotal, Sai Sree Laya Chukkapalli, and Anupam Joshi	
Toward Green Federated Learning	409
Minsu Kim and Walid Saad	

Part I
Security and Privacy

Trustworthiness, Privacy, and Security in Federated Learning



Sisi Zhou, Lijun Xiao, Yufeng Xiao, and Meikang Qiu

1 Introduction

GDPR is a landmark bill in global personal data security legislation, introduced by the European Union in 2018. The bill is divided into 99 Articles, granting data subjects the right to agree, access, correct, be forgotten, restrict processing, refuse, and automate self-determination, among other normative data rights. The law on personal data privacy was introduced in Germany as early as 1970, followed by data protection laws introduced by Switzerland, the United States, and others. According to incomplete statistics, it covers over 140 countries worldwide. The GDPR introduced by the European Union has the greatest impact. Part of the global data security legislation is shown in Fig. 1.

Governments around the world have established various forms of legal and compliant guidelines and norms for data usage, gradually forming a global awareness of data security. The era of barbaric use and arbitrary sharing of data has come to an end. However, in order to ensure the orderly circulation of data elements under the premise of legal compliance and ensuring the rights and interests of all parties, it undoubtedly increases the cost for data users to obtain and store user personal

S. Zhou (✉) · Y. Xiao
School of Computer Science and Engineering, Hunan University of Science and Technology,
Xiangtan, China
e-mail: sisizhou@mail.hnust.edu.cn; hnxiaoyf@hnust.edu.cn

L. Xiao
College of Information Engineering, Shanghai Maritime University, Shanghai, China
e-mail: ljxiaoxy@126.com

M. Qiu
School of Computer and Cyber Sciences, Augusta University, Augusta, GA, USA
e-mail: qiumeikang@ieee.org

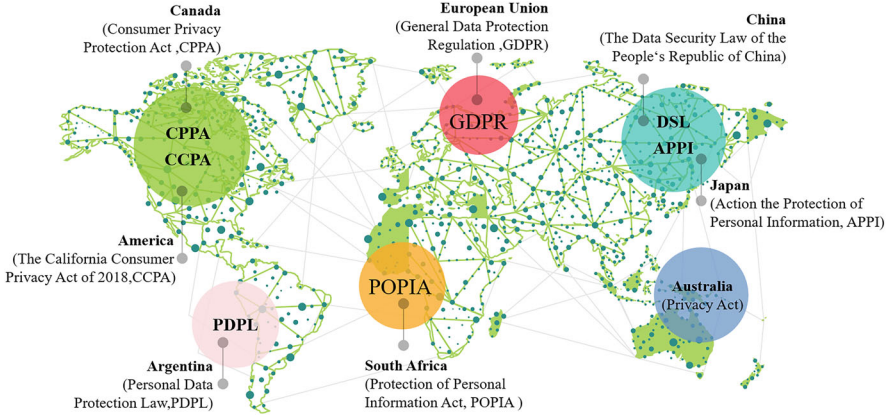


Fig. 1 Global Partial Data Security Legislation

information through Internet technology. Exploring feasible models for data element circulation has important practical significance [1].

In the context of the GDPR [2], legal compliance issues have been raised for a range of emerging technologies including *Internet of Things* (IoT), *artificial intelligence* (AI), and big data analytics, and also blockchains [3]. Especially in the field of machine learning where data is regarded as the core, industry, academia, and users are paying more attention to data privacy protection, believing that private data should not be exposed or uploaded to central servers. Based on this, the federated learning [4], proposed by Google in 2016, effectively solves the contradiction between data privacy and data sharing on decentralized devices by the feature of “data available but not visible.” This framework belongs to the distributed training model and has two roles, namely the participating device and the central server. Nodes update the global model locally, upload model updates (i.e., local gradients), and do not upload private data. The central server collects updates and integrates them to form an updated model. Due to this privacy characteristic of FL, it has received widespread attention and rapid development in recent years.

However, under the federated learning framework, the server needs to collect update information from multiple clients for aggregation operations and needs to broadcast new global models to these clients, which requires high network bandwidth [5, 6] and computing capability [7–9]. In addition, centralized servers may have behaviors that favor certain clients and distort the global model, and malicious central servers may also disrupt the model and even collect client privacy from updates. Therefore, the single-point failure of the center server and the fairness and security issues of the federated learning have attracted extensive attention [10].

In recent years, blockchain has innovative integrated technologies such as cryptography, distributed consensus algorithms, and P2P networks to build a decentralized trust environment. For example, B. Thuraisingham and M. Qiu et al. proposed a novel secure data sharing algorithm through untrusted clouds with

blockchain-enhanced key management method [11] and several blockchain-enabled service optimizations in supply chain digital twin [12], which can be widely used in computer vision [13] and cybersecurity [14, 15] areas.

Z. Tian et al. [16] proposed a lightweight blockchain-based secure digital evidence framework, which combines a mixed block structure with an optimized name-based practical Byzantine fault tolerance consensus mechanism. The multisignature technique is adopted for evidence submission and retrieval, ensuring the traceability and privacy of evidence. H. Qiu et al. [17] proposed a dynamic trust system based on blockchain that will provide a dynamic and scalable communication architecture for IoT networks. W. Pan et al. [18] proposed to use blockchain technology to improve the *Asset-Backed Securitization* (ABS) financial business system. By analyzing the current situation and existing issues of ABS business in China, the problems to be solved are summarized, and a design scheme of ABS business system based on blockchain technology is proposed.

Blockchain, as a distributed, decentralized, immutable, and transparent technology, provides new possibilities for the development of federated learning, providing new possibilities for the development of federated learning [19]. Hou et al. [20] summarized and compared the infrastructure and application scenarios of some existing blockchain and federated learning fusion frameworks. Wahab et al. [21] conducted research on federated learning, covering aspects such as federated learning architecture, privacy protection, communication efficiency, etc., which also includes some integrated architectures of blockchain and federated learning. Kim et al. [22] and Qu et al. [23] point out that decentralized blockchain is commonly used in the fusion framework of blockchain and federated learning to replace the central server in traditional federated learning frameworks, in order to solve the problems of single-point trust and failure caused by the central server.

From recent research work on the integration of blockchain technology and federated learning technology, research in this field mainly focuses on the BC-empowered FL framework itself and its application prospects in the field of artificial intelligence. The BC-empowered FL is broadly used in diverse fields, including industrial Internet, intelligent transportation, smart healthcare, and wireless network infrastructure. It has shown tremendous success, and several impactful solutions have been proposed, such as BlockFL [22] and Deepchain [24]. In the literature, there are many surveys about blockchain and federated learning, respectively. However, these surveys mainly focus on the practical applications, challenging issues, and technical solutions of blockchain or federated learning. Despite their comprehensiveness in either area, they rarely explain the potential of using blockchain for federated learning. In other words, none of them systematically studies the BC-empowered FL providing an overview of the current research trends and future directions.

The rest of this chapter is organized as follows. In Sect. 2, federated learning framework is given. Section 3 explains the Blockchain-empowered Federated Learning framework. Next, we summarize the future research directions in Sect. 4.

2 Federated Learning

With the increasing number of IoT devices, the amount of generated data is also growing rapidly. This presents significant privacy concerns, particularly when using centralized AI solutions. Centralized AI solutions are less suitable for modern IoT applications due to the heterogeneous nature of IoT devices' data, resources, and their distribution across multiple geographical locations [25]. To address these privacy and security issues associated with centralized AI solutions, the architectures of FL have been proposed.

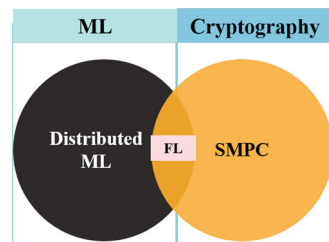
Federated learning technology, as a computational framework for multiparty joint modeling, interacts with model parameters through security mechanisms to achieve collaborative training effects [26]. Federated learning belongs to distributed machine learning methods and belongs to the category of privacy computing. It aims to avoid sharing the private local data of IoT devices with a central server. Instead, in FL, each IoT device participating in the FL process trains the machine learning (ML) model locally using its own data. Only the learned model parameters are shared with the central server for global model aggregation. This approach improves privacy, saves communication resources, and enhances the robustness of the training process compared to centralized ML.

In detail, privacy computing in FL-related technologies includes machine learning [27], distributed machine learning [28], cryptography [29], differential privacy [30], secure multiparty computing [31], and other different technologies in addition to federated learning. This framework can promote distributed collaborative learning without disclosing the original training data. It is a promising framework to give play to the value of data fusion in the increasingly strict global environment of data security and privacy protection. The relationship between federated learning and related technologies is shown in Fig. 2.

Taking the training of a *deep neural network* (DNN) model as an example, we assume that a central server delegates a DNN task. Under the federated learning framework, all training data is retained at the user end. The basic process of FL with K clients is shown in Fig. 3.

The central server first assigns an initialized model to K clients, and each client of the client set completes local training using the newly received model and sends the updated model to the central server. From then on, a round of updates is completed. Repeat these processes until the model converges or reaches a specific number of

Fig. 2 The relationship between federated learning and the related technologies



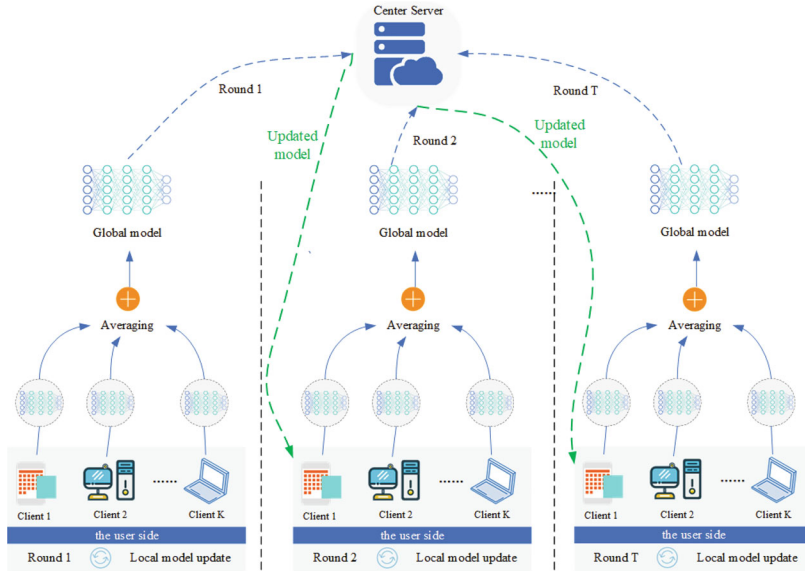


Fig. 3 The basic process of FL

rounds. Since staff only submit models, their raw data will not be shared with the central server, so their privacy can be protected to some extent. Federated learning runs round by round as follows:

- **Client selection.** A new round starts. The coordinator selects some clients from sample devices based on certain criteria, e.g., historical activities, model quality, network bandwidth, and computation capabilities. The criteria are designed in advance to get the federated learning system rid of malicious client devices.
- **Model selection and initialization.** This step determines which type of ML is the most suitable and whether it is a pretrained or new model that needs to be trained from scratch. Besides that, the model parameters are initialized W_G^0 , and the model is broadcast to the participating clients K to start the FL training rounds T , where W_G^0 is the initial global model, K is the total number of selected clients, and T refers to the total number of global training rounds.
- **Local model training.** Each selected client receives the shared global machine learning model from the coordinator and executes a model training program to update the local machine learning model taking local data as input. More specifically, in the first round, every client k receives the initial global model W_G^0 and begins the local training on its data $D_k \in D$, where $D = \{D_1 \cup D_2 \cup \dots \cup D_k\}$ and D_k is the local dataset of the k -th client. Subsequent training rounds will use the latest global model for local training. Hence, in every global round $t \in T$, each client k calculates the local model update w_k^t by minimizing a loss function. After the local training is finished for a round, the local learned parameters will be sent to the central entity for aggregation.

- **Global model aggregation.** The coordinator aggregates the models or updates from the client devices. To improve efficiency, the coordinator may stop the aggregation once enough client devices have submitted the models or updates. From the client’s perspective, they may slightly adjust the models before sending them to the coordinator for data privacy concerns.
- **Global model update.** The coordinator updates the shared global model based on the models or updates from the client devices participating in the current round. The model update algorithm can be a simple average of the received models or updates. The design of the model update algorithm is critical for the convergence speed and accuracy of the final output model.
- **Convergence checking.** The coordinator calculates the model difference between two consecutive rounds. If the difference is smaller than a predefined threshold, the procedure ends. Otherwise, it goes back to the step of client selection. The local model training, global model aggregation, and update are repeated until the number of training rounds is reached or the predefined accuracy threshold is achieved. The server stops the training at this point, and the trained global model is broadcast to all clients.

By following these steps, FL enables collaborative learning among distributed IoT devices without compromising data privacy. It allows IoT devices to contribute to the training process while keeping their data local, thereby addressing the challenges of trustworthiness, privacy, security, communication efficiency, and robustness in modern IoT applications.

2.1 The Architectures of FL

In this section, we will explore different architectures of FL and how the parameter exchange occurs between the clients and the FL coordinator.

A. Centralized FL

The architecture of centralized FL relies on a centralized server to initiate the FL process and aggregate the global model. It represents the traditional FL approach and follows the same steps outlined earlier. Centralized FL addresses privacy concerns by avoiding the transmission of sensitive client data to the central server. However, it suffers from robustness issues due to a single point of failure. The architecture of centralized FL is depicted in Fig. 4.

Although FL has some privacy protection effects, large-scale data collection and processing on powerful cloud-based servers will bring risks of single point of failure and serious data leakage. From the GDPR perspective, Truong et al. [32] pointed out that centralized data processing and management have limited transparency and sources for the system, which may lead to a lack of trust among end users and difficulty in complying with GDPR.

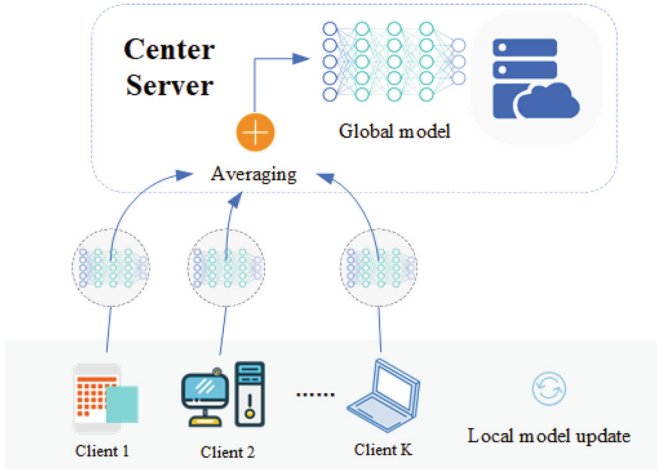


Fig. 4 The architecture of centralized FL

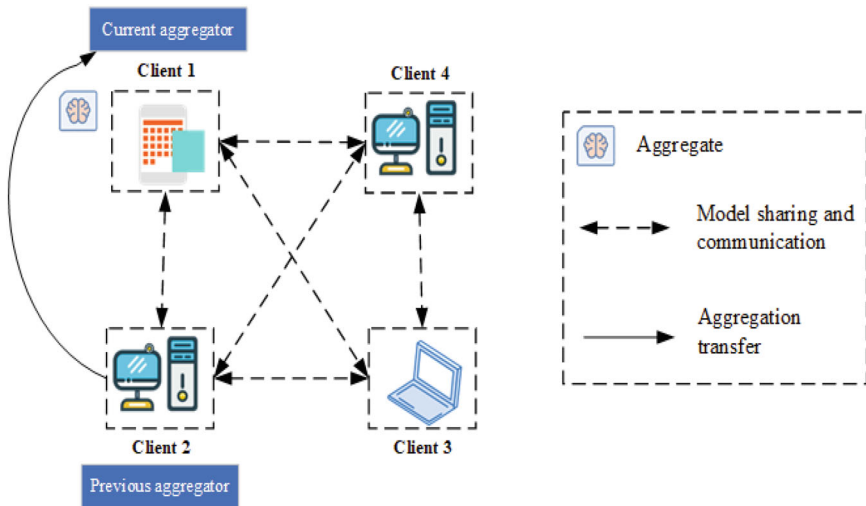


Fig. 5 The architecture of decentralized FL

B. Decentralized FL

In contrast to a centralized architecture, in the decentralized federated learning architecture, also known as a peer-to-peer network architecture [33], participating client devices can communicate directly with each other without the need for a third-party server. The architecture of decentralized FL is depicted in Fig. 5.

Under the decentralized FL framework, all clients connect in a peer-to-peer(P2P) or mutual communication manner to exchange local model updates and aggregate the global model. Centralized FL architectures are unsuitable for environments

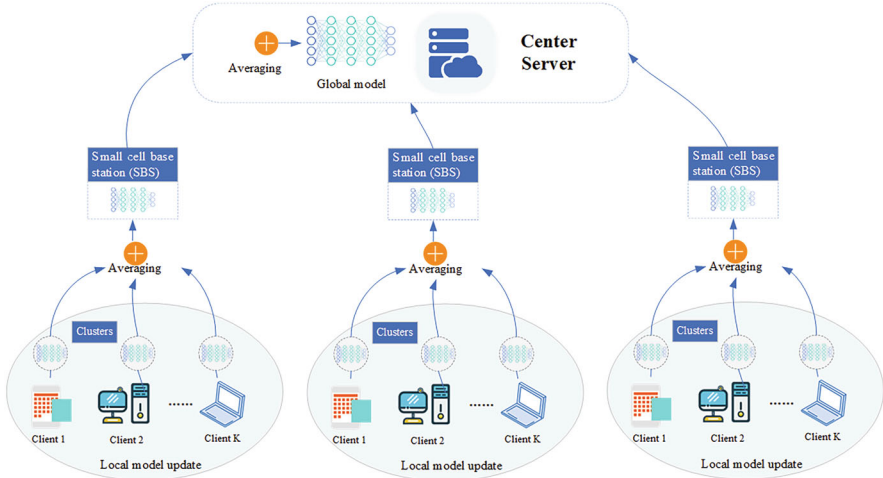


Fig. 6 The architecture of hierarchical FL

where it is likely that communication between participating clients and the central entity will be unstable. In such cases, it is recommended to use the decentralized federated learning architecture. For instance, clients in a P2P network can communicate via blockchain ledger to store their local updates and aggregate the global model in a trusted and secure way.

C. Hierarchical FL

In the architecture of Hierarchical FL, *mobile users* (MUs) are clustered into groups (clusters) based on their locations, and each cluster is assigned to a *small cell base station* (SBS). MUs train models on their local data and send their local updates to the SBS. This process is repeated for a specific number of iterations. Afterward, all SBSs send the aggregated local updates to the *mobile base station* (MBS) to aggregate the global model. The architecture of hierarchical FL is depicted in Fig. 6.

Abad et al. [34] showed that the architecture of hierarchical FL helps reduce the communication latency, speeding up the training of the global model.

D. Collaborative FL

In centralized FL, there are scenarios where a subset of IoT devices may fail to send their local model updates to the central server. This can occur due to various reasons such as low energy requirements of the devices, limited communication resources, or transmission delays. To address these challenges, a collaborative federated learning architecture has been proposed by Chen et al. [35]. In the collaborative FL framework, it is not necessary for all devices to be connected directly to the central server. Instead, some IoT devices connect to the central server, while others connect to neighboring devices based on their proximity. This creates a mesh-like network structure. While the proposed framework helps overcome some limitations of centralized FL, it also has its own limitations. These include slower convergence

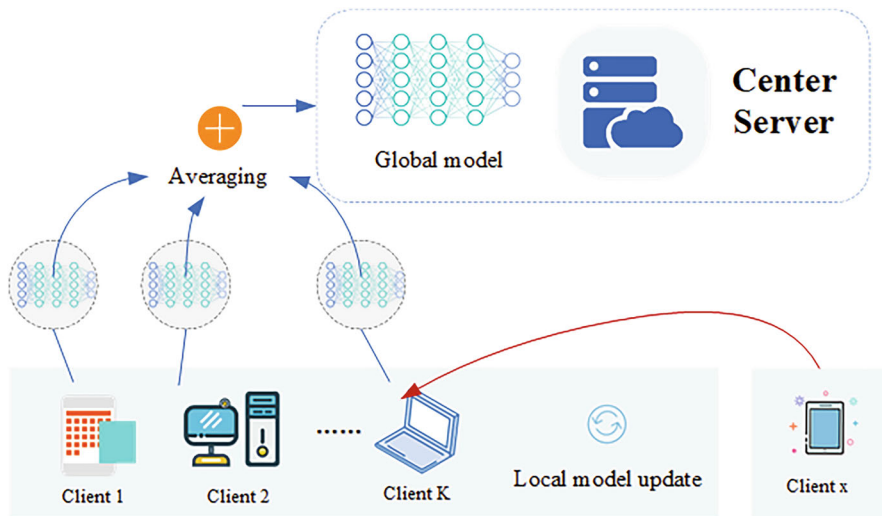


Fig. 7 The architecture of collaborative FL

speed compared to centralized FL, potential impact of imperfect communication between devices on the training process, and different ML convergence for each group of associated devices. Additionally, the centralized server still represents a single point of failure. The architecture of collaborative FL is depicted in Fig. 7.

E. Dispersed FL

Dispersed FL is a distributed federated learning framework that enables global model learning in two stages. In the first stage, subglobal models are aggregated within different groups, each consisting of closely located devices. In the second stage, the global model is computed by aggregating the subglobal models either in a centralized or distributed manner. Dispersed FL can be categorized into two types: centralized dispersed FL and distributed dispersed FL. However, according to Khan et al. [36], dispersed FL still has limitations concerning client privacy and *non-Independent and Identically Distributed* (non-IID) data. The architecture of dispersed FL is depicted in Fig. 8.

From a system architecture perspective, both federated learning and traditional distributed learning consist of servers and multiple distributed nodes, exhibiting high similarities. However, compared to traditional distributed learning, federated learning has its own characteristics in terms of data, communication, and system composition. Specifically, the optimization problem is implicit in federated learning as federated optimization, drawing a connection (and contrast) with distributed optimization. Federated optimization has several key properties that differentiate it from a typical distributed optimization problem:

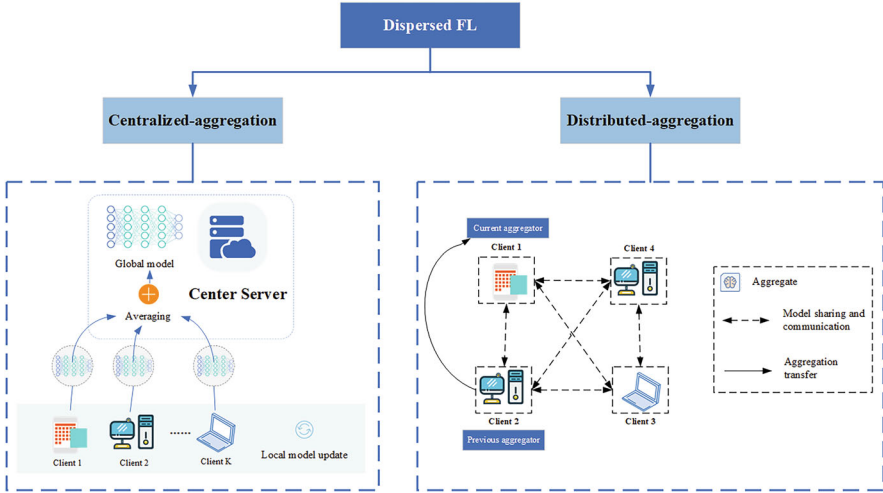


Fig. 8 The architecture of dispersed FL

- **Non-IID.** The training data on a given client is typically based on the usage of the mobile device by a particular user, and hence any particular user's local dataset will not be representative of the population distribution.
- **Unbalanced.** Similarly, some users will make much heavier use of the service or app than others, leading to varying amounts of local training data.
- **Massively distributed.** We expect the number of clients participating in an optimization to be much larger than the average number of examples per client.
- **Limited communication.** Mobile devices are frequently offline or on slow or expensive connections.

2.2 Scale and Data Partitions in FL

FL systems can be categorized into either cross-device or cross-silo, based on the number of participating clients and their data volume. The cross-silo and cross-device structures are depicted in Fig. 9.

Cross-device It is an FL approach that involves a large number of clients with limited data size. Thus, the number of the involved devices ranges from millions to billions. Examples of cross-device FL systems include IoT devices and smartphones. Due to the volume of devices, selecting the most qualified devices (i.e., with enough computational, communication, and energy resources) to participate effectively in the FL is important.

Cross-silo Unlike the cross-device FL approach, the cross-silo approach has a small number of clients with a large data volume. Such clients may be data centers

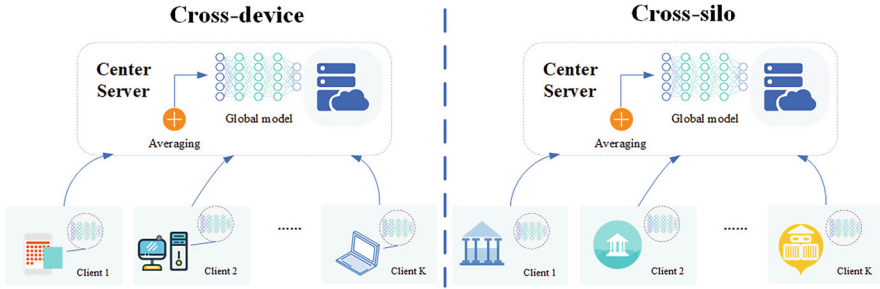


Fig. 9 The cross-silo and cross-device structures

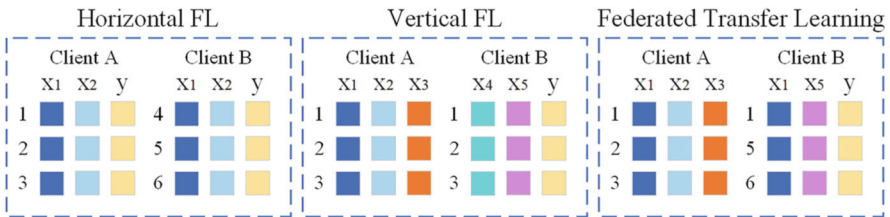


Fig. 10 The three categories of FL

or organizations. For instance, the Amazon product recommendation model learns via the contribution of numerous data centers, each using its local data.

When designing FL models, it is crucial to take into account data distributions and partitions. Based on the sample and feature spaces, FL can be categorized into the three categories, namely horizontal FL, vertical FL, and federated transfer learning. The three categories of FL are depicted in Fig. 10.

Horizontal FL is dubbed sample-based FL, where the clients that participated in the FL process have different data samples with the same feature space. Therefore, participating clients can use the same ML to be trained locally due to the same feature space. For instance, next word prediction models learn from datasets with different sample spaces and the same feature space to predict the next word.

Vertical FL is known as feature-based FL, where clients' datasets have the same sample space with different feature spaces. For example, in IoT applications, the ML model can be shared among different entities such as banks and e-commerce companies that serve clients in the same city (the same sample space). At the same time, different users will collect various features. For example, a bank may contain a set of features that differs from the features collected by the e-commerce application for the same client (different feature spaces). Furthermore, the bank and the e-commerce company can cooperatively train ML models using the Vertical FL scheme to predict the personalized loans based on the online shopping activities of clients.

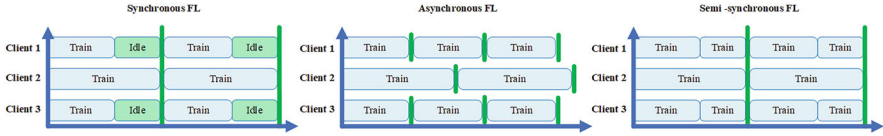


Fig. 11 Three synchronization schemes

Federated transfer learning is a combination between vertical FL and horizontal FL, where the clients have datasets with different sample spaces and different feature spaces. Furthermore, federated transfer learning is vital in transforming different sample spaces' features into the same representation. For instance, federated transfer learning can be used in disease diagnosis applications where the global model training depends on clients from different countries (sample space) and has different medical tests (feature space).

2.3 Aggregation Time Schemes

FL approaches can be classified into three synchronization schemes based on the timing of global model aggregation, as depicted in Fig. 11.

The details of these three schemes are as follows:

Synchronous FL does not consider the heterogeneity of edge devices, which may have varying computational and energy resources. As a result, it does not effectively utilize the resources of participating devices, as high-computational devices remain idle until other devices complete their local training. Additionally, in real scenarios, some devices may join midway through the process, while others may fail to submit their local updates. The speed of rounds in synchronous FL is limited by the slowest device, leading to a “straggler effect” that hampers efficient processing. Although synchronous FL consumes low communication resources, it exhibits slow learning convergence.

In contrast, asynchronous FL does not have a specific synchronization point. Participating devices can submit their local updates and download new versions of the computed global model whenever they complete local training. Asynchronous FL approaches have higher convergence but consume more communication resources compared to synchronous FL. Feng et al. [37] proposed another asynchronous FL approach to improve scalability and efficiency while addressing poisoning attacks targeting asynchronous FL.

Semisynchronous FL is considered as a middle ground solution between the synchronous and asynchronous FL methods. In a semisynchronous FL, the participating devices are permitted to train the ML locally up to a certain synchronization point where the global model is calculated. As a result, this method lowers communication costs and makes better use of the resources of participating devices. In general, the semisynchronous FL approach has been proposed to balance communication costs

and resource usage. Stripelis et al. [38] proposed a semisynchronous FL approach that accelerates the model convergence while reducing communication cost. Also, it improves resource utilization by eliminating the idle time of high-computational devices and involving the local update of low-computational devices in the global model computation.

2.4 Federated Optimization

A. The Federated Averaging Algorithm

Federated averaging (FedAvg) [4], proposed by McMahan, is a communication-efficient algorithm for distributed training with an enormous number of clients. In FedAvg, clients keep their data locally for privacy protection; a central parameter server is used to communicate between clients. The complete pseudocode of FedAvg is given in Algorithm 1.

Algorithm 1 FedAvg

```

1 Server executes:
2 initialize  $W_G^0$ 
3 for each round  $t = 1, 2, \dots$  do
4    $m \leftarrow \max(C \cdot K, 1)$ 
5    $S_t \leftarrow$  (random set of  $m$  clients)
6 for each client  $k \in S_t$  in parallel do
7    $w_{t+1}^k \leftarrow$  ClientUpdate( $k, w_t$ )
8    $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$ 
9 ClientUpdate( $k, w$ ): // Run on client  $k$ 
10   $\mathcal{B} \leftarrow$  (split  $\mathcal{P}_k$  into  $o$  batches of size  $B$ )
11  for each local epoch  $i$  from 1 to  $T$  do
12    for batch  $b \in \mathcal{B}$  do
13       $w \leftarrow w - \eta \nabla \ell(w; b)$ 
14  return  $w$  to server

```

The clients K are indexed by k , B is the local minibatch size, T is the number of local epochs, and η is the learning rate. We write to indicate that the full local dataset is treated as a single minibatch.

B. Adaptive Federated Optimization

In this section, some adaptive federated optimizations are introduced, namely, FedAdagrad, FedYogi, and FedAdam. Their pseudocode is given in Algorithm 2.

The parameter τ controls the algorithms' degree of adaptivity, with smaller values of τ representing higher degrees of adaptivity. Note that the server updates of our methods are invariant to fixed multiplicative changes to the client learning rate η for appropriately chosen.

Algorithm 2 FedAdagrad, FedYogi, and FedAdam

1 **Initialization:** $x_0, v_{-1} \geq \tau^2$, decay parameters $\beta_1, \beta_2 \in [0, 1)$

2 **for** $t = 0, \dots, T - 1$ **do**

3 Sample subset S of clients

4 $x_{i,0}^t = x_t$

5 **for each client** $i \in S$ **in parallel do**

6 **for** $k = 0, \dots, K - 1$ **do**

7 Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(w_{i,k}^t)$

8 $x_{i,k+1}^t = x_{i,k}^t - \eta 1 g_{i,k}^t$

9 $\Delta_i^t = x_{i,K}^t - x_t$

10 $\Delta_t = \frac{1}{|S|} \sum_{i \in S} \Delta_i^t$

11 $m_t = \beta_1 m_{t-1} + (1 - \beta_1) \Delta_t$

12 $v_t = v_{t-1} + \Delta_t^2$ (**FedAdagrad**)

13 $v_t = v_{t-1} + (1 - \beta_2) \Delta_t^2 \text{sign}(v_{t-1} - \Delta_t^2)$ (**FedYogi**)

14 $v_t = \beta_2 v_{t-1} + (1 - \beta_2) \Delta_t^2$ (**FedAdam**)

15 $x_{t+1} = x_t + \eta \frac{m_t}{\sqrt{v_t + \tau}}$

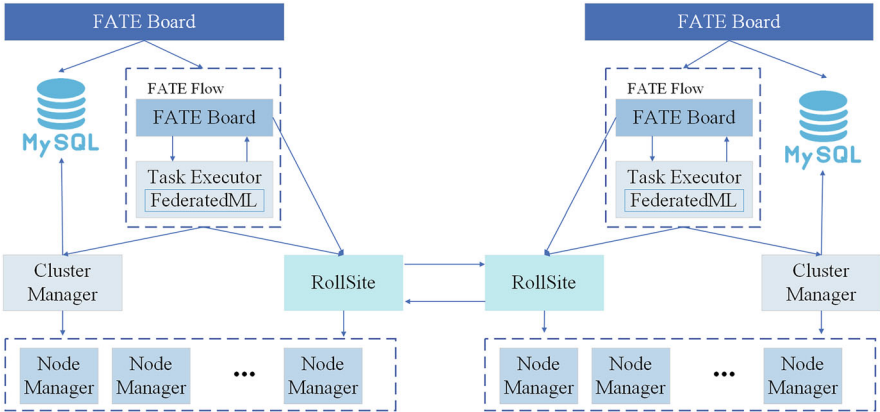


Fig. 12 The architecture of FATE

2.5 Federated Learning Platforms

In order to mitigate the risk of privacy breaches and enhance the security of sensitive data, both industry and academia have introduced several open-source frameworks for federated learning, such as: *Federated AI Technology Enabler* (FATE) [39], PaddleFL [40], and *federated learning natural language processing* (FedNLP) [41].

FATE helps to learn and understand the theory of federated learning. Its architecture mainly consists of two components: offline training and online prediction. The architecture of FATE is depicted in Fig. 12.

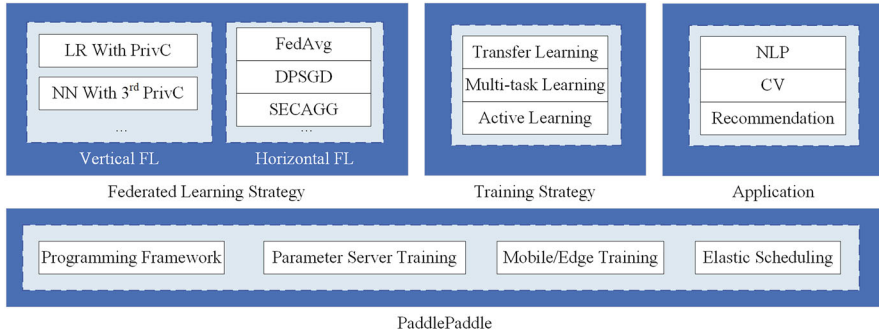


Fig. 13 The architecture of PaddleFL

FATE Flow is the learning task pipeline management module responsible for job scheduling in federated learning. Federation serves as the data communication module in the federated network, facilitating message transmission among different functional units. Proxy acts as the network communication module and handles routing functionality. Meta-service functions as the cluster metadata service module. MySQL serves as the foundational component for both the meta-service and FATE-Flow, storing system data and work logs. FATE Serving provides the online federated prediction module, offering federated online inference capabilities. FATE-Board is the module for visualizing the federated learning process. Egg and Roll respectively serve as the distributed computing processor management module and computation result aggregation module, responsible for data computation and storage.

PaddleFL supports two types of strategies: horizontal federated learning and vertical federated learning. For horizontal federated learning, it primarily supports strategies such as FedAvg, DPSGD, and SECAGG. PaddleFL adopts the underlying programming model of the Paddle training framework. By combining Paddle's parameter server functionality, it enables the deployment of federated learning systems in Kubernetes clusters. In terms of training strategies, PaddleFL supports multitask learning, transfer learning, active learning, and other training techniques. Please note that the translation provided above may require further review and adjustment based on the specific technical terminology and context used in the PaddleFL documentation. The architecture of PaddleFL is depicted in Fig. 13.

FedNLP, developed by Lin et al. at the University of Southern California, is the first research-oriented open-source federated learning framework for *natural language processing* (NLP). It consists primarily of three layers: the application layer, the algorithm layer, and the infrastructure layer. The architecture of FedNLP is depicted in Fig. 14.

At the application layer, FedNLP offers a wide range of predefined NLP tasks and datasets, allowing researchers to easily apply federated learning to various NLP problems such as text classification, named entity recognition, sentiment analysis, and machine translation. The algorithm layer of FedNLP incorporates state-of-

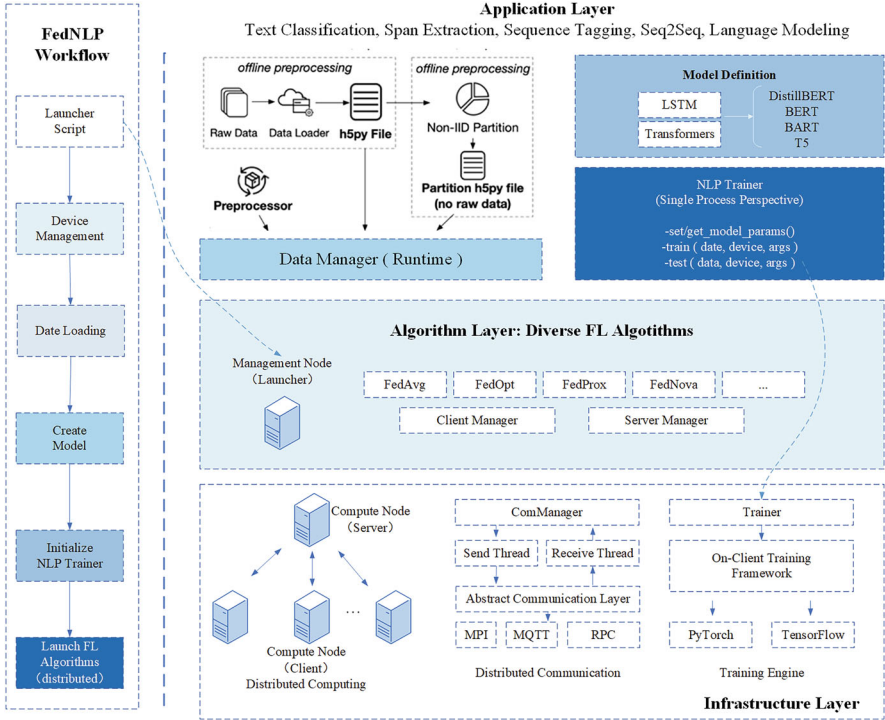


Fig. 14 The architecture of FedNLP

the-art federated learning algorithms specifically designed for NLP tasks. These algorithms include federated optimization techniques, secure aggregation methods, and privacy-preserving mechanisms to ensure efficient and secure collaboration among distributed NLP models. The infrastructure layer of FedNLP leverages the underlying federated learning frameworks to enable seamless deployment and scalability. It supports distributed computing platforms such as TensorFlow and PyTorch, allowing researchers to leverage the power of distributed systems and parallel computing for federated NLP.

2.6 Solutions of the Security and Privacy in FL

Several techniques aim to design a secure and privacy-preserved FL architecture. However, designing a secure and privacy-preserving FL while maintaining model accuracy and optimizing computational and communication resources is challenging. The most common approaches designed to produce secure FL can be summarized as follows:

Homomorphic encryption (HE): It is a method of exchanging encrypted model parameters to protect the privacy of users and reduce the risk of data leakage. However, HE requires a trade-off between accuracy and privacy. Some researchers proposed federated learning privacy preservation approaches based on HE to encrypt the local gradients. Further, it employed distributed selective stochastic gradient descent to minimize the computation cost.

Secure multiparty computation (SMC): It is a multiparty proof-of-zero-knowledge security mechanism. Each party has no knowledge of the other and only has direct exposure to its related data. Because of its complicated calculations, achieving zero-knowledge proof is usually impossible. Therefore, whenever the required security constraints are satisfied, the partial zero-knowledge proof is acceptable. Moreover, SMC relies on cryptographic techniques for securing client updates in the FL. SMC outperforms traditional cryptographic mechanisms by encrypting only the model parameters rather than a large volume of data, making it preferable in FL applications. The main issue with SMC is the trade-off between privacy and deficiency, as SMC execution takes time and has a negative impact on FL training. As a result, lightweight SMC solutions for FL are still required.

Differential privacy (DP): It is an approach that aims to protect the client's sensitive data from privacy leakage in FL. DP adds a small amount of noise to the local model parameters to make it difficult for attackers to extract personal information about the participants. However, the DP approaches reduce the privacy leakage risk, but there is still a trade-off between the amount of added noise and the overall model accuracy. Furthermore, adding more noise will prolong the model convergence time. To avoid privacy leakage of FL in IoV environments, some researchers proposed Local Differential Privacy (LDP) mechanisms. This LDP mechanism reduced communication costs while preventing adversaries from recreating exact training data from vehicle gradients.

Anonymization: Anonymization techniques are used to protect privacy by removing personally identifiable and sensitive data while maintaining data utility. Three techniques are commonly used to achieve the data anonymization: k-anonymity, l-diversity, and t-closeness. These techniques have been applied in recent peer-reviewed research.

Blockchain: It is a distributed ledger technology characterized by immutability, transparency, reliability, trustworthiness, auditability, and accountability features. These promoting features make the blockchain a more convenient for defending against FL attacks. Due to its security and traceability features, blockchain is an excellent choice for serving as a decentralized coordinator in FL. Some researchers proposed an algorithm to protect FL from model poisoning attacks. The algorithm is implemented and run in the smart contract on the blockchain. Unlike other algorithms that use data sample size or reputation to verify the quality of local updates, this algorithm uses accuracy as a metric to verify the accuracy of local updates before aggregating the global model. Similarly, some researchers introduced blockchain-enabled federated learning frameworks for securing the control of urban traffic flow. The authors revealed

that blockchain has been used to allow decentralized federated learning, in which model updates can be verified by miners to avoid fraudulent updates and so mitigate the impact of data poisoning attacks.

3 The Blockchain-Empowered FL Framework

3.1 Blockchain Technology

Blockchain, conceptualized by a pseudonymous creator Satoshi Nakamoto in 2008 [32], is a decentralized shared ledger (DSL) that combines data blocks in a chronological order into a specific data structure in a chain manner and ensures tamper resistance and unforgeability through cryptography. In a broad sense, it is an entirely new infrastructure and distributed computing paradigm. Taking the Bitcoin block, for example, the blockchain structure is shown in Fig. 15.

The data layer encapsulates the blockchain’s underlying encryption technology and data storage method.

The network layer involves the distributed peer-to-peer network and the transport mechanisms required to connect and operate among network nodes.

The consensus layer includes various consensus mechanisms. They are combined with incentive mechanisms to achieve data consistency among nodes.

The contract layer is a programmable implementation of blockchain technology.

The application layer draws support from the underlying technology to implement various application scenarios and cases.

As an analogy to the OSI 7-layer model, the basic architecture of blockchain can be divided into six layers: data, network, consensus, incentive, contract, and application. The basic architecture of blockchain is shown in Fig. 16.

Regardless of the concept, blockchain has the characteristics of distributed, tamper proof, smart contracts, and encryption, as follows:

- **Distributed structure.** It not only implements P2P mode for data transmission, but also enables mutual verification of node information, thereby forming

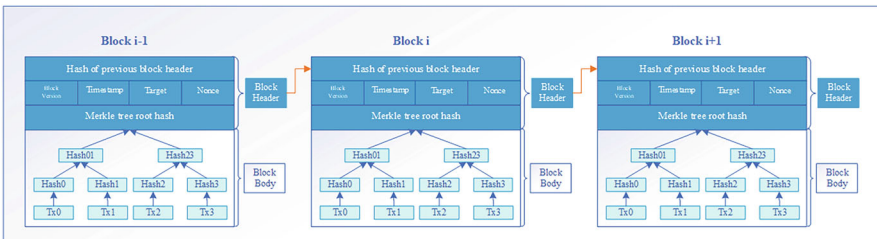


Fig. 15 The structure of the blockchain

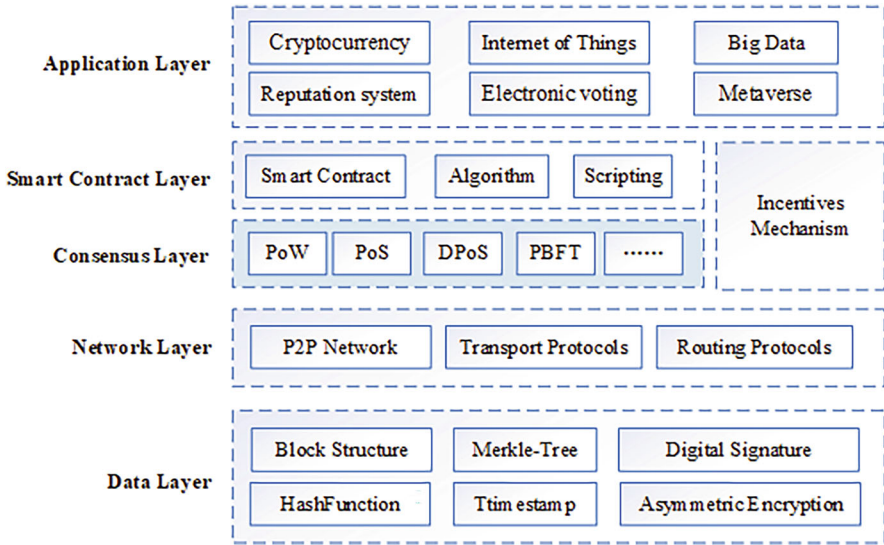


Fig. 16 The basic architecture of blockchain

consensus on a certain information. Consensus information is stored by each user on local carriers (such as mobile phones or computers), greatly increasing the cost of tampering with the new information.

- **Immutability.** Because every node on the blockchain records the transaction information, if you want to change the transaction information, taking Bitcoin as an example, you must use more than 51% of the blockchain's computing power to complete it. The more users in blockchain, the more users need to obtain consent when tampering with data information, and the greater the difficulty.
- **Smart Contract.** Smart contracts utilize computer algorithms to transform traditional contracts into automated execution techniques triggered by intelligent recognition. When the contract conditions agreed upon by all parties are met, the automatic execution instruction of the smart contract is triggered, and the contract will be irreversibly and automatically fulfilled.
- **Encryption.** The most critical technology of blockchain should be digital encryption technology, which utilizes a digital encryption algorithm. This encryption algorithm is generally divided into symmetric encryption algorithm and asymmetric encryption algorithm, and asymmetric encryption algorithm is mainly used in blockchain.

With immutability and trustworthiness, blockchain has developed into an enabling technology and has built a credible digital environment, widely used in broader fields and deeper scenarios, such as Liang et al. [42] proposed a blockchain-based homomorphic encryption for IP circuit protection recopyright transactions. Chen et al. [43] introduced the medical data privacy protection method based on blockchain technology data privacy protection. Liang et al. [44] proposed a