Future Generation Computer Systems 150 (2024) 272–293

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Review article

Model aggregation techniques in federated learning: A comprehensive survey

Pian Qi [a], Diletta Chiaro [a], Antonella Guzzo [b], Michele Ianni [b], Giancarlo Fortino [b], Francesco Piccialli [a,*]

[a] Department of Mathematics and Applications "R. Caccioppoli", University of Naples Federico II, Naples, Italy
[b] Department of Computer Science, Modeling, Electronic and System Engineering (DIMES), University of Calabria, Calabria, Italy

ARTICLE INFO

Article history:
Received 21 March 2023
Received in revised form 19 July 2023
Accepted 3 September 2023
Available online 11 September 2023

Keywords:
Model aggregation
Model fusion
Federated learning
Distributed machine learning
Machine learning
Artificial intelligence

ABSTRACT

Federated learning (FL) is a distributed machine learning (ML) approach that enables models to be trained on client devices while ensuring the privacy of user data. Model aggregation, also known as model fusion, plays a vital role in FL. It involves combining locally generated models from client devices into a single global model while maintaining user data privacy. However, the accuracy and reliability of the resulting global model depend on the aggregation method chosen, making the selection of an appropriate method crucial. Initially, the simple averaging of model weights was the most commonly used method. However, due to its limitations in handling low-quality or malicious models, alternative techniques have been explored. As FL gains popularity in various domains, it is crucial to have a comprehensive understanding of the available model aggregation techniques and their respective strengths and limitations. However, there is currently a significant gap in the literature when it comes to systematic and comprehensive reviews of these techniques. To address this gap, this paper presents a systematic literature review encompassing 201 studies on model aggregation in FL. The focus is on summarizing the proposed techniques and the ones currently applied for model fusion. This survey serves as a valuable resource for researchers to enhance and develop new aggregation techniques, as well as for practitioners to select the most appropriate method for their FL applications.

Contents

* Corresponding author.
E-mail address: francesco.piccialli@unina.it (F. Piccialli).

https://doi.org/10.1016/j.future.2023.09.008

1. Introduction

Since its very first introduction, machine learning (ML) has garnered immense popularity across various fields due to its remarkable problem-solving capabilities. ML's capacity to uncover concealed information and identify non-linear patterns in data has provided significant benefits to scientific research and real-life applications. Consequently, it is increasingly being utilized in industries such as finance, healthcare, and business [1]. However, while ML brings substantial advantages, its data-driven nature has raised concerns regarding privacy and security [2]. This is primarily because the potential for data leakage or illicit data collection poses security risks within the ML space. In traditional artificial intelligence (AI) technologies, such as ML itself, the data used for training models are usually concentrated in the data center. Large datasets have enabled incredible advances in ML. Once these data centers storing a large amount of data are attacked, the consequences are immeasurable. Furthermore, data are frequently personal or proprietary and are not intended to be shared, making privacy a critical concern and roadblock to centralized data gathering and model training. Based on the above-made premise, with the purpose of protecting privacy, a distributed training approach, federated learning (FL) came into being.

Federated learning represents a novel paradigm in ML that aims to facilitate the training of high-quality models by coordinating multiple clients or devices, all while preserving the privacy of their respective local datasets. The foundational framework for FL was initially proposed by the Google team [3], and since then, it has gained increasing popularity among researchers. This is primarily due to its inherent ability to provide enhanced privacy protection compared to traditional ML approaches.

As research interest in FL has grown, several variations of the initial server–client model have emerged. These include fully decentralized FL, where only clients are involved [4], as well as FL built on blockchain technology [5], among others. In the most common setup, FL consists of a central server and multiple clients. The server's role is to coordinate the collaborative training process, while the clients act as individual participants. These clients can range from small devices with computational capabilities (such as IoT devices, mobile phones, or computers) to large organizations or institutions. Importantly, each client retains its own local dataset and does not share it with other participants, which forms the fundamental premise of FL.

Fig. 1 shows an overview of the general FL framework. In each communication round, the following steps occur: first, each client downloads the latest model from the server to initialize their local model. Second, clients employ their own local datasets to perform training and update their respective local models. Subsequently, the clients send their updated models back to the server. Finally, the server performs model aggregation on the received local models, resulting in a new global model. These steps are repeated until communication ceases.

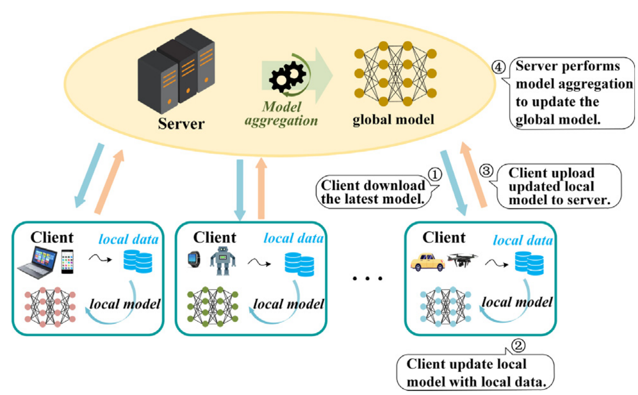


Fig. 1. Overview of the general FL framework. During the tth communication round, each client downloads the latest global model from the server for initialization ① and uses its own local dataset for iterative training ②. Then, the updated weights are uploaded to the server ③ which performs model aggregation ④ to generate a new global model. These steps are repeated for each communication round.

Compared to traditional ML approaches where all data is collected centrally, FL ensures that local data remains on the client devices. This characteristic provides enhanced privacy and security guarantees. While FL can be considered a form of distributed ML, it also presents unique challenges and issues to address, such as the statistical heterogeneity of data and device heterogeneity [6]. Statistical heterogeneity refers to the fact that the data distributed among clients often have different distributions, deviating from the assumption of independent and identically distributed (IID) data. This heterogeneity can lead to challenges such as reduced model accuracy and slower convergence speed [7]. Device heterogeneity, on the other hand, pertains to variations in computing capabilities and transmission speeds among different clients, which can impact the operational efficiency of FL to varying degrees.

Undoubtedly, FL has gained significant traction in various practical applications, particularly in the realm of edge computing [8]. The proliferation of edge computing has been enabled by the enhanced computational capabilities of modern devices like mobile phones, wearables, drones, and autonomous vehicles [9]. By integrating FL technology into the edge network, local data can be stored on terminal devices, local training can be conducted, and the data can subsequently be aggregated and updated by a central server. This approach addresses the imperative need for data privacy on devices while effectively coordinating a large number of remote devices.

A review of relevant literature on FL reveals that a substantial portion of the research has focused on applying FL to diverse domains such as healthcare [10], smart transportation [11], wireless networks [12], and the Internet of Things (IoT) [13]. Several systematic federated frameworks have been implemented, including the mobile device federated system [14].

Researchers have dedicated considerable efforts to studying FL, encompassing a broad range of topics, including data distribution patterns [15], advancements in model aggregation [16], and practical applications of FL [17]. Since the introduction of the FedAvg algorithm, which is based on weighted averaging, by McMahan et al. [18], an increasing number of studies have focused on exploring aggregation strategies in FL. Thus, the motivation behind this paper is to conduct a systematic and comprehensive survey of existing model aggregation methods in the field of FL, aiming to provide valuable references for researchers and practitioners. Currently, only a few surveys are available, such as [19,20], which discuss the crucial issue of model aggregation in FL. However, these surveys cover only a limited number of research works and techniques. Recognizing the significance and effectiveness of model aggregation techniques, we conducted a systematic literature review of model fusion methodologies employed in FL from 2017 to 2023. The objective of this study is to identify the scope, trends, and methods used in the field of model fusion, ultimately enhancing our understanding of this domain and providing a comprehensive taxonomy of relevant methods. To achieve these objectives, we formulated the following research questions:

- RQ1: What are the model aggregation methods utilized by the research community in FL, and how do they compare in terms of their advantages and disadvantages?
- RQ2: What are the primary ML/DL algorithms employed in the field of FL, and how do they impact the model aggregation phase?
- RQ3: Can existing aggregation methodologies be grouped to provide a comprehensive and accurate taxonomy?
- RQ4: What are the most popular journals publishing literature on model aggregation?
- RQ5: What are the main application domains of FL, and how do they relate to model aggregation techniques? Are model aggregation techniques specific to certain domains?
- RQ6: What are the main challenges encountered during the model aggregation phase?

1.1. The contributions of the survey

In recent years, the remarkable growth of FL and its applications has led to an upsurge in related surveys and reviews. Table 1 provides a summary of recent surveys that have garnered significant citations in the field of FL. Notably, the papers [6,19] offer introductions to the fundamental concepts of FL and discuss the challenges it faces. Meanwhile, [20,21] focus on security and privacy concerns in FL, summarizing potential risks and vulnerabilities in the context of FL. And, [22] conducts a survey on personalization in FL, highlighting the impact of non-IID data on federated models. Furthermore, [23] delves into the exploration of non-IID data in FL. The authors of [24] present a summary of relevant consensus protocols, platforms, and real-world cases in FL. Moreover, [9,25–29] provide reviews on FL research applications, covering areas such as healthcare, edge networks, 5G, and IoT.

However, to the best of our knowledge, while there are numerous reviews available that focus on various aspects of FL, such as privacy and security issues, non-IID data, and applications in different domains, the topic of model aggregation has not received the attention it deserves. Despite the claim by [27] to be a review of model aggregation, it merely provides a brief introduction to several classic aggregation methods without conducting a comprehensive taxonomy or summary.

This highlights the gap in the existing literature, further emphasizing the need for a systematic and comprehensive review specifically focusing on model aggregation techniques in FL. The present study aims to fill this gap and provide a thorough examination of the various methodologies employed in model aggregation within the context of FL.

Key contributions of the survey are enlisted as follows:

1. The paper provides a comprehensive and systematic review of model aggregation techniques in FL. It includes a detailed taxonomy and literature survey, offering a comprehensive overview of recent work in the field.
2. In addition to discussing model aggregation strategies, the paper explores the impact of data heterogeneity on FL models. It also introduces real-world applications of FL, highlighting the relationship between model aggregation and different application domains.
3. The paper identifies current challenges in FL, such as communication bottlenecks and privacy and security concerns. Furthermore, it examines potential future trends and developments in the field, providing insights into the direction of research and advancements in FL.

1.2. The structure of the survey

This survey is organized as follows: firstly, in Section 2, we outline the systematic approach employed to retrieve research work on model aggregation in FL, while Section 3 provides a comprehensive introduction to FL, covering its fundamental concepts and principles. Section 4 presents a detailed survey and taxonomy of model aggregation methods in FL. In Section 5, we explore application scenarios related to FL from the perspective of model aggregation. Section 6 delves into the challenges currently faced by FL, including statistical heterogeneity of data, communication bottlenecks, and security threats. We also propose potential research directions to address these challenges and advance the field of FL. Fig. 2 provides a visual representation of the structure of this survey, illustrating the flow and organization of the different sections.

2. Research methodology

To evaluate the applications and impact of aggregation methods in FL, a systematic literature review was conducted. This involved following a systematic approach to identify and analyze pertinent studies in a specific area of interest [30], ensuring that the review process was scientific, transparent, and reproducible. Expert-defined guidelines were utilized to ensure a comprehensive and effective systematic literature review. Specifically, the guidelines proposed by Siddaway et al. [31] and the PRISMA statement by Moher et al. [32] were studied and adhered to. The Siddaway et al. proposal outlines an eight-step process for conducting a systematic literature review, including formulating research questions, defining inclusion/exclusion criteria, searching the literature, screening studies, extracting data, assessing

Table 1
Existing surveys on FL topics and our contributions.

Paper	Topic	Key contribution	Statistical heterogeneity	Communication	Security and privacy	Model aggregation	Taxonomy
[6]	FL concept	Discussed the properties and challenges of FL; provided a survey of classical results, and summarized open issues.	✓	✓	✓	✗	✗
[19]	FL concept	Introduced the current work of FL from 5 aspects, sorted out the current challenges and future research directions of FL.	✗	✓	✓	✓	✓
[21]	Security and privacy in FL	Analyzed the privacy leakage risk in FL from 5 aspects, summarized the existing methods and future research directions.	✗	✓	✓	✗	✓
[20]	Security and privacy in FL	Discussed threats in FL and evaluated related mitigation techniques.	✗	✓	✓	✓	✓
[22]	Personalization in FL	Explained statistical heterogeneity due to non-IID data; highlighted the need for model personalization.	✓	✓	✓	✗	✗
[23]	Non-IID data in FL	Analyzed the impact of non-IID data on parametric and nonparametric ML models in FL; reviewed current challenges.	✓	✓	✓	✗	✓
[24]	Platforms, protocols, and applications	Summarized the most relevant protocols, platforms, and real-life use cases for FL; outlined the main challenges.	✗	✓	✓	✗	✗
[25]	FL and healthcare	Presented the motivation and requirements for using FL in smart healthcare; reviewed emerging applications of FL in key medical areas.	✓	✓	✓	✗	✓
[9]	FL and edge networks	Introduced the applications, challenges, and future directions of FL in mobile edge networks optimization.	✓	✓	✓	✗	✗
[26]	FL and 5G	Discussed possible applications and key technical challenges in 5G networks.	✗	✓	✓	✗	✗
[28]	FL and IoT	Introduced the recent progress of FL in IoT applications.	✗	✓	✓	✓	✓
[27]	FL model aggregation	Current aggregation techniques and challenges in FL are discussed.	✓	✓	✓	✓	✗
Ours	FL model aggregation	A systematic survey on model aggregation in FL, from scientific literature retrieval to detailed taxonomy of aggregation methods; also explores hot issues in FL, including non-IID data, communication, security, and privacy.	✓	✓	✓	✓	✓

study quality, analyzing and synthesizing results, and reporting findings. On the other hand, the PRISMA statement provides a checklist for reporting systematic reviews and meta-analyses, covering items such as the title, abstract, introduction, methods, results, discussion, and funding.

In the subsequent sections, we present the search strategy and selection criteria employed to identify relevant studies for this systematic literature review. The research questions, as previously formulated in the introduction section, served as a guide throughout the review process.

2.1. Search strategy

At the beginning of the literature search, we decide to utilize the *Scopus* and *Web of Science* databases, given their comprehensive coverage of papers. As mentioned earlier, the concept of FL was first introduced by the Google team in October 2016 [3], and the FedAvg algorithm was defined by McMahan et al. in 2017 [18]. These works established the foundation for the concept of FL as intended in this survey. Therefore, our literature search was limited to publications between 2017 and June 2023, encompassing formal publications and preprints, while excluding non-English articles.

To ensure comprehensive coverage of relevant literature, the primary search term "federated learning" was used to search the titles, abstracts, and keywords of the literature. Furthermore, in order to account for the possibility that some researchers may use the terms "fusion" and "aggregation" interchangeably, the secondary search term "model aggregation" OR "model fusion" was employed to search all sections of the literature. It is important to note that despite using these search keywords, it is possible that not every relevant publication was captured without omission. This is because, in the early stages of FL development, some researchers preferred to name their proposed frameworks instead of explicitly using the term "model aggregation". Examples of

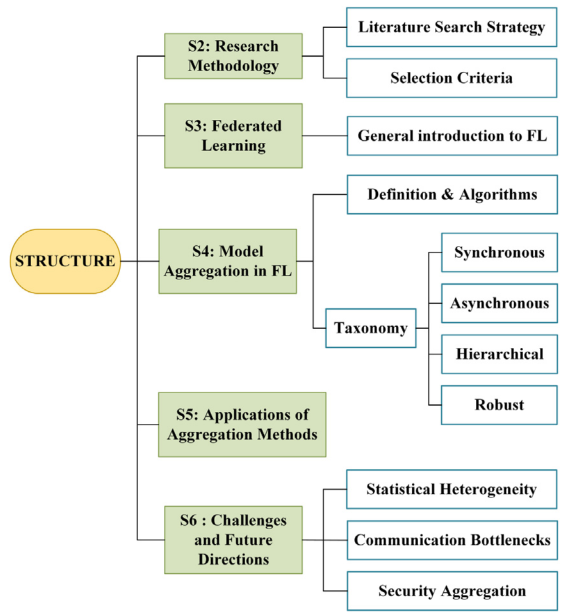


Fig. 2. The structure of the survey. The survey is organized into several sections, starting with an overview of the methodology employed for conducting the study. Following that, it introduces the concept of aggregation methods in FL, providing readers with essential background information. Subsequently, a comprehensive and detailed taxonomy of the various aggregation methods proposed thus far is presented. The survey also features a discussion on the application of FL aggregation methods in diverse data types and scenarios. Lastly, the survey concludes with a summary of future research directions in this field.

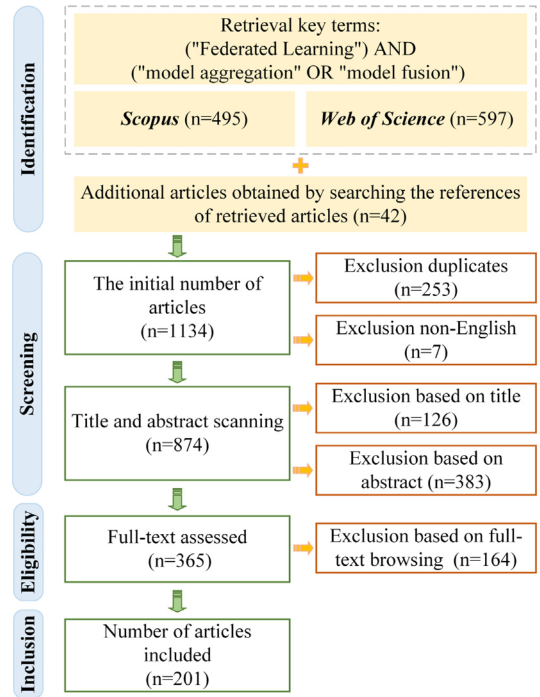


Fig. 3. The systematic literature review conducted in this study follows a research article retrieval methodology that consists of four stages: identification, screening and eligibility, and inclusion. In the identification stage, the appropriate keywords were determined to retrieve research articles relevant to the research topic under investigation. In the screening and eligibility stages, specific criteria were established to screen and exclude literature that did not meet the research needs and objectives. Finally, in the inclusion step, the literature that met the criteria and requirements of the study was determined and included in this paper.

such frameworks include FedProx [7], FedNova [33], Scaffold [34], among others. To mitigate the risk of missing articles that did not explicitly mention the keywords, we also searched the references of the retrieved papers for further relevant publications.

2.2. Selection criteria

The literature search began by conducting a comprehensive search using key terms, which resulted in finding 495 articles on *Scopus* and 597 articles on *Web of Science*; duplicates were excluded. Additionally, it was acknowledged that some of the initially retrieved literature may not be relevant to the research topic. To address this, a set of criteria was developed for screening the literature. Review articles and surveys were excluded from the analysis, as the focus was specifically on experimental articles. Next, the titles and abstracts of the articles were scanned to exclude those that did not mention model aggregation design or research. Finally, the full text of the remaining articles was reviewed, and articles that only used existing methods in the model aggregation stage without proposing any new ones were excluded. Following this screening process, a total of 933 articles were eliminated, resulting in a final set of 201 articles that were deemed suitable for inclusion in the survey. The literature selection process is depicted in Fig. 3, comprising four stages: identification, screening, eligibility, and inclusion.

Initial statistics have been compiled for the 201 selected articles. Fig. 4 presents the distribution of research articles focusing on model aggregation techniques in FL from 2017 to June 2023. It indicates that the majority of published articles (75%, i.e., 151 articles) have been released in the past two years (2021 to the present), indicating the active and current nature of research in model aggregation in FL. Out of the screened papers, 113 are from IEEE Computer Society, encompassing various field journals such as IEEE Conference on Computer Vision and Pattern Recognition,

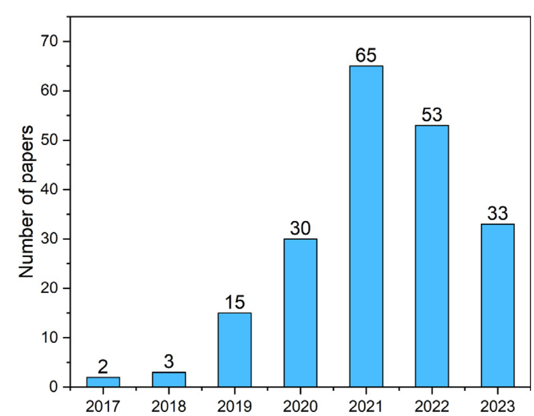


Fig. 4. Year-wise distribution of papers on the subject area of model aggregation in FL.

IEEE International Conference on Computer Vision, IEEE Internet of Things Journal, IEEE Transactions on Wireless Communications, among others. Furthermore, there are 20 papers from ACM (including the International Conference on Machine Learning and ACM Conference on Embedded Networked Sensor Systems), 20 papers from Elsevier, 11 papers from Springer, 5 papers from MIT Press (including Advances in Neural Information Processing Systems), and 32 papers from other journals. Table 2 lists the Journal-wise distribution of papers.

Table 2
Journal-wise distribution of papers on the subject area of model aggregation in FL.

Journal		No.	Percentage
IEEE		113	56.22%
	IoT-J	10	
	TWC	6	
	CVPR	5	
	ICC	5	
	IJCNN	4	
	Trans. Wirel. Commun.	6	
	Trans. Neural Netw. Learn. Syst.	3	
	J. Sel. Areas Commun.	4	
	Trans. Parallel Distrib. Syst.	6	
	Trans. Ind. Informat.	5	
	Others	59	
ACM		20	9.95%
	ICML	11	
	Others	9	
Elsevier		20	9.95%
Springer		11	5.47%
MIT Press		5	2.49%
	NeurIPS	5	
Others		32	15.92%

3. Federated learning

In this section, we provide a brief introduction to FL, including its principles, features, and categories.

3.1. Principles of FL

FL generally refers to a distributed ML process deployed on multiple clients. The process involves N clients $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ indexed by k, each with its own local dataset $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$, which is kept locally, and data cannot be shared between clients or collected by a third party. Typically, a server coordinates different clients and their training. FL involves three key steps:

(1) Initialization: at communication round t, the clients download the latest model w^t from the server for initialization;

(2) Local training: each client $\mathcal{C}_k$ performs iterative training based on its own local dataset $\mathcal{D}_k$ and hyperparameter η. The local model w_k^t is updated to w_k^{t+1} after certain training epochs according to $w_k^{t+1} \longleftarrow w_k^t(\eta, D_k)$, and then send to the server.

(3) Model aggregation: the server performs model aggregation on received local models, and updates the global model $w_{glob}^{t+1} \longleftarrow Agg(w_k^{t+1};\ k \in [1, \ldots, N])$.

In this way, FL enables multiple clients to collaborate on training a model without sharing their data, which is especially useful for privacy-sensitive applications.

3.2. Features of FL

This section covers unique features that set FL apart from traditional centralized ML approaches. These distinguishing features include aspects such as data and device heterogeneity, as well as the specific topology of the federated scenario.

3.2.1. Data distribution

In FL, one of the major challenges is the presence of statistical heterogeneity in the data, also referred to as non-IID-ness. Unlike traditional centralized ML approaches that assume independent and identically distributed (IID) data structures, FL data often exhibits statistical heterogeneity due to variations in client distribution, data generators, or other factors. This can result in local models that are biased towards the distribution of the local dataset, which can cause a decline in the performance of the aggregated model.

Three primary sources of non-IID-ness can be identified:

- Label distribution skew. Label distribution skew refers to the variation in the distribution of labels, denoted as $P(y)$, across different clients. In typical FL experimental setups, datasets are sorted by labels and assigned to clients in a round-robin fashion. For example, in studies such as [35–37], the MNIST dataset is assigned to clients, where each client is allocated only two out of the total ten labels. This means that client i may have data labeled as “1” and “2”, while client j has handwritten data labeled as ”3“ and ”4”, with no label overlap between clients. This allocation method is commonly used and straightforward. Furthermore, researchers commonly employ the *Dirichlet* distribution $Dir(\beta)$, which is based on the Bayesian prior distribution, to simulate the distribution differences in FL data [16,38,39]. In this context, the concentration parameter β plays a crucial role in controlling the non-IIDness of the data. A smaller β value corresponds to higher data heterogeneity. By manipulating β, researchers can flexibly generate various data distributions to investigate the effects of non-IIDness on FL models.
- Feature distribution skew. Feature distribution skew refers to the variation in the distribution of features, denoted as $P(x)$ or input data, across different clients in FL. This variation can lead to discrepancies in the representation of data for the same class across different clients. To address feature distribution skew, researchers have explored various methods. Some approaches involve adding noise to the dataset or using federated datasets specifically designed to mitigate this issue [40]. The Federated Extended MNIST (FEMNIST) dataset is another example of a dataset created to address feature distribution skew in FL [41]. FEMNIST includes 62 different handwritten characters, comprising 10 numbers, 26 lowercase letters, and 26 uppercase letters. It partitions the data from the Extended MNIST (EMNIST) dataset, which consists of characters written by different authors with varying styles and degrees of sloppiness, thus skewing the feature distribution.
- Quantity skew. Quantity skew refers to the unequal amount of data that different clients have in a FL setting. It is not feasible to ensure that every client has the same amount of data, and in practice, the data is often randomly partitioned among clients. For instance, in [42], the CIFAR-10 dataset is partitioned among clients with different amounts of training data, ranging from 400 to 1600 examples per client. The amount of data each client has can significantly affect the performance of the model, and this skew must be taken into account when designing and evaluating FL algorithms.

3.2.2. Cross-device and cross-silo

FL can be categorized into cross-device and cross-silo based on the difference in client size. Cross-device FL is often employed in distributed mobile networks [14], where numerous similar devices act as clients. Real-world examples of cross-device FL include federated systems based on the Internet of Vehicles, where each vehicle independently collects its own driving data (such as captured images and current coordinates). On the other hand, cross-silo FL involves large organizations or institutions as clients, with fewer clients compared to cross-device FL. In this case, different companies may collaborate to build a federated system, where each company maintains local data and performs model training based on their respective databases.

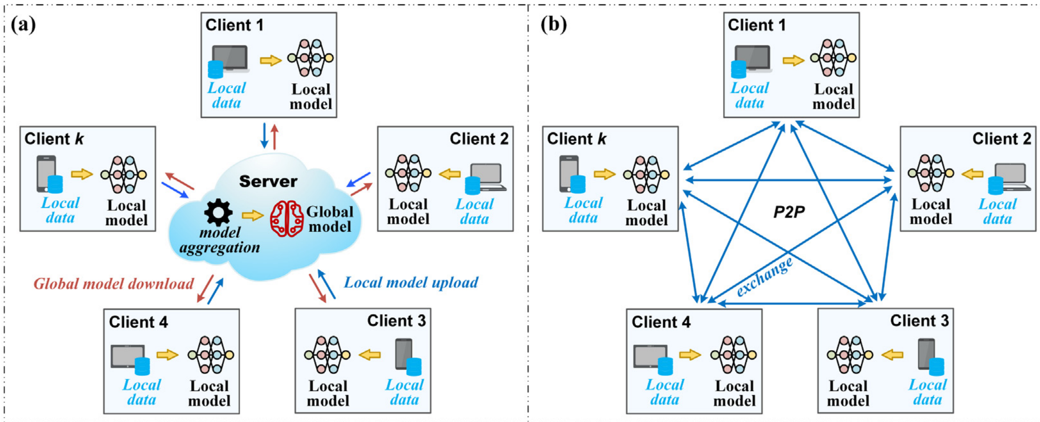


Fig. 5. The two primary network structures for FL: centralized FL and decentralized FL. In a centralized FL system (a), there is a server positioned at the center, forming a star network structure. Multiple clients connect to this central server for model aggregation and synchronization. On the other hand, in a decentralized FL system (b), there is no central server. Instead, clients directly communicate with each other in a peer-to-peer (P2P) manner, creating a mesh network structure. Decentralized FL is specifically designed to mitigate the presence of untrusted servers and offers advantages such as increased resilience to network failures and communication delays.

3.2.3. Network structure

During the initial stages of FL design, a common depiction involved a central server positioned at the center, surrounded by multiple clients. In this setup, the central server was responsible for the model aggregation phase, while each client performed local model training using its own data. This configuration, known as centralized FL, typically employed a star topology. As illustrated in Fig. 5(a), the central server was situated at the center, with all other clients connected to it. In contrast, there exists a distinct federated configuration referred to as decentralized FL [43], which adopts a mesh topology (see Fig. 5(b)). In this structure, there is no central server. Each client in the system conducts local model training based on its private data. Subsequently, clients select other clients to communicate with, and local models are exchanged or fused in a peer-to-peer (P2P) manner. This decentralized approach facilitates model aggregation and updates without relying on a central server, thereby mitigating the presence of untrusted servers.

3.2.4. Data partition

FL can be classified into three categories based on the distribution of feature space and sample space: horizontal FL, vertical FL, and federated transfer learning.

- Horizontal Federated Learning. Horizontal FL, also known as sample-based FL, involves clients sharing the same feature space but different sample spaces. This means that data from different clients may be sampled from different objects, such as A, B, C, etc., but all share the same characteristics, such as color, as shown in Fig. 6(a). For example, a study using FL to detect COVID-19 infection [44] used chest CT images as training data for each client, which were sampled from people of different ages and genders but had the same feature space.
- Vertical Federated Learning. Vertical FL, also referred to as feature-based FL, is a situation where the training data used by all clients participating in training have the same sample space but different feature spaces. This means that the data is sampled from the same object (object A), but different features (different colors) are distributed among different clients, as depicted in Fig. 6(b). The primary goal of vertical FL is to align overlapping data samples between different clients, and it is commonly used in scenarios where non-competing companies or organizations, such as financial institutions, e-commerce platforms, and advertising companies with different data characteristics, collaborate to train a shared learning model. By leveraging vertical FL, these organizations can collectively develop a model that facilitates personalized shopping experiences for online shoppers within the same region, sharing a common sample space. For instance, financial institutions can provide prepaid funds to users when they make purchases on online shopping platforms, while advertising companies can offer personalized suggestions based on users' purchasing behaviors. This collaborative model allows for the integration of different data characteristics from various sources, enabling the delivery of enhanced personalized services to users.
- Federated transfer learning. Federated transfer learning involves data from different participants that not only differ in samples but also in feature spaces. Fig. 6(c) illustrates the schematic diagram of federated transfer learning, where local data comes from different objects (object A, B, C...) with different features (different colors). In this approach, common representations between different feature spaces are typically learned from a limited set of common samples. These common representations are then applied to the sample prediction task with only one-sided features [45]. An example of this is FedHealth [46], which is a framework for researching wearable healthcare using federated transfer learning. In this framework, the cloud server trains the cloud model based on the basic dataset. After obtaining the cloud model, transfer learning is performed in combination with the client's local data to establish a personalized local model.

4. Model aggregation in FL: Towards a taxonomy

In this section, we first introduce the definition of model aggregation. Then, we show some well-known aggregation algorithms. Finally, we categorize model aggregation based on different forms of aggregation.

4.1. Definition of model aggregation

In FL, model aggregation refers to summarizing model parameters from all parties in each round of communication to form an updated global model. Privacy protection is achieved by aggregating model parameters instead of raw training data.

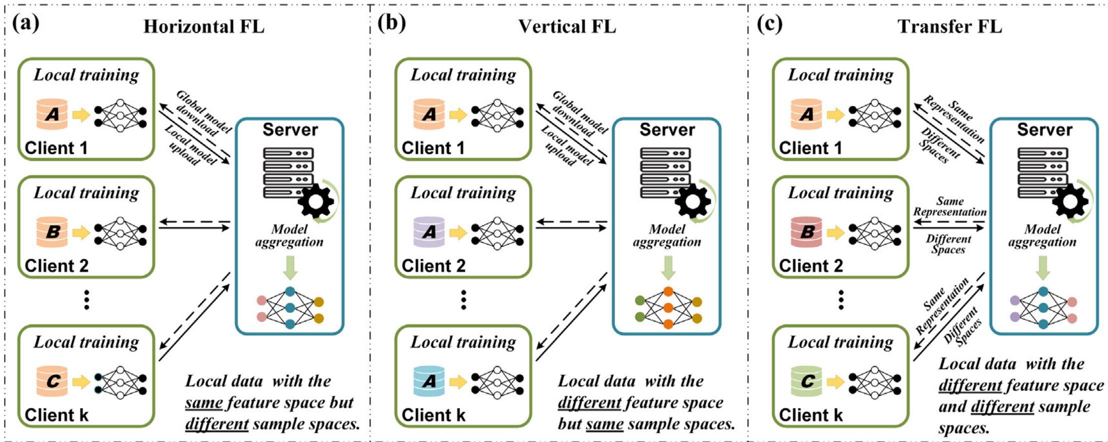


Fig. 6. Data partition in FL: (a) Horizontal FL, (b) Vertical FL and (c) Federated transfer learning.

Model aggregation can be divided into two types: parameter-based aggregation and output-based aggregation, based on the objects being aggregated. In parameter-based aggregation, trainable parameters of the local learning model, such as the weight parameters and gradients of the deep neural network, are the objects of aggregation. In each round of communication, local models are shared in parameters/gradients after iterative training based on their local dataset [47,48].

Output-based aggregation focuses on aggregating the representations of the model, such as the output logits or compressed sketches [39,49]. A notable example of output-based aggregation is the Fedmask approach introduced by Li et al. [50]. Fedmask takes into account the limited computing resources of mobile devices in FL. In this approach, each device learns a binary mask, and the server aggregates the overlapping binary masks to enhance the computational efficiency of training. By aggregating the binary masks, the computational burden on individual devices is reduced while still achieving effective model training.

When considering the form of aggregation, in centralized FL, the model aggregation is conducted by the central server. The central server is responsible for collecting and merging the models from individual clients to create an aggregated model. In contrast, in decentralized FL, model aggregation occurs through peer-to-peer (P2P) communication. Typically, one party initiates the communication and collaborates with neighboring clients or collaborators to aggregate their respective models. This decentralized approach involves the exchange and fusion of models directly between the participating clients or collaborators, without relying on a central server for aggregation.

4.2. Main algorithms for model aggregation

There have been numerous traditional aggregation algorithms proposed to address fundamental challenges in FL, particularly related to communication overhead and data privacy concerns. These algorithms are often incorporated into various federated frameworks as foundational computing approaches. Below, we provide an introduction to several of these algorithms.

- **FedAvg**
One of the earliest and most commonly used methods for FL is FedAvg [18]. In FedAvg, a group of clients is randomly selected at each round of training for aggregation. During the aggregation process, the parameters of each client are weighted and averaged to produce a global model, where the weight factor is the proportion of the client's data volume. Note that, in the FedAvg's implementation it could be added more computation to each client by iterating the local update multiple times before the averaging step. During the tth communication round, FedAvg works as $w_{glob}^{t+1} \leftarrow \sum_{k\in S_t} \frac{n_k}{n} w_k^{t+1}$, where $k \in S_t$ represents the members of the selected set of clients; $\frac{n_k}{n}$ refers to the weight factor, which is equal to the ratio of the data volume of client k to the total data volume. Finally, w_k^{t+1} represents the updated model of client k after local training, and w_{glob}^{t+1} represents the aggregated global model.
- **FedProx**
FedProx [7] is an enhancement to the FedAvg method aimed at mitigating the problem of local optimization inherent in SGD-based approaches. The authors posit that performing numerous local iterative training steps in FedAvg may cause each client to prioritize achieving its local objective rather than the global goal, leading to suboptimal convergence or model divergence. In FedProx, a proximal term is incorporated into the objective function to regulate the influence of local models and ensure convergence guarantees. Such term, defined as $\frac{\mu}{2}\left\|w_k^t - w_{glob}^t\right\|^2$, is the l^2-norm of the local model and global model. Here, $\mu \geq 0$ is the penalty constant of the proximal term, and FedProx is equal to FedAvg when $\mu = 0$. The local model is pulled towards the global model through the constraints of the proximal term. The subsequent model aggregation and global model update follow the same process as FedAvg.
- **FedNova**
The FedNova algorithm, proposed by Wang et al. in [33], enhances the model aggregation phase of the FedAvg algorithm to address non-IID-ness. The algorithm introduces a technique to normalize and scale the local updates from each client based on its local iteration number, before updating the global model. The update rule for FedNova is defined as follows:

$$w^{t+1} \leftarrow w_{glob}^t - \tau_{eff} \sum_{k\in S_t} \frac{n_k}{n} \cdot \eta d_k^t$$

$$\text{where} \quad d_k^t = G_k^t a_k^t / \left\|a_k^t\right\|^1$$

where τ_{eff} denotes the effective iteration step. The local updates d_k^t are calculated by normalizing the gradients with a non-negative vector a_k^t and its l^1-norm. In the case of using vanilla SGD as the local solver, a_k^t is a unit vector. The stack of all stochastic gradients received from client k at round t is denoted as G_k^t.

- **Scaffold**
To address the client-drift problem resulting from data heterogeneity, the Scaffold algorithm [34] suggests employing control variable technology, specifically variance reduction, in the local update process. This algorithm incorporates client control variable c_k and server control variable c, with the server control variable being the average value of the control variables from all clients. The disparity between local and global control variables is utilized to rectify gradient updates during local training. The update procedure for the control variables in Scaffold is as follows:

$$c_k^+ \leftarrow c_k - c + \frac{1}{e\eta_l}(w_{glob}^t - w_k^t)$$

$$c^+ \leftarrow c + \frac{1}{|N|}\sum_{k \in S_t}(c_k^+ - c_k)$$

where "$^+$" denotes update and η_l is learning rate. e is the number of local update steps, and N represents the total number of clients.
- **MOON**
MOON [51] aims to minimize the discrepancy between local models and the global model by incorporating a model contrastive loss as a regularization term. The model contrastive loss serves as a metric for quantifying the dissimilarity between the local model and the global model. Its definition is as follows:

$$l_{con} \leftarrow -log\frac{exp(sim(w_k^t, w_{glob}^t)/\tau)}{exp(sim(w_k^t, w_{glob}^t)/\tau) + exp(sim(w_k^t, w_k^{prev})/\tau)}$$

The model-contrastive loss used in MOON is defined as a function of three terms: the global model w_{glob}^t, the previous round model w_k^{prev}, and the current local model w_k^t. Also, *sim* represents the cosine similarity, and temperature parameter τ is used to control the sharpness of the loss function. The purpose of the model-contrastive loss is to encourage the local models to be close to the global model while avoiding overfitting the current local data.
- **Zeno**
To prevent Byzantine faults in FL, the Zeno algorithm uses a stochastic zero-order oracle to score each candidate client. The scoring is based on the loss function, and is used to select high-scoring clients for aggregation. More specifically, Zeno defines the Stochastic Descendant Score (SDS) for any gradient update. The SDS score measures how much the candidate client contributes to reducing the loss of the global model, while penalizing large updates that may cause instability. The high-scoring clients are then selected for aggregation, and their updates are combined to form a new global model. This approach allows Zeno to select reliable clients and improve the robustness of the learned global model against Byzantine behavior.
- **Per-FedAvg**
Per-FedAvg [52] combines model-agnostic meta-learning (MAML) with FL to produce personalized local models. MAML first trains the initial parameters of the model, and then uses a small amount of data to perform one or more gradient descents on the new task, enabling the model to achieve good performance. For more information on MAML, please refer to [53]. In Per-FedAvg, the local model is updated using one-step gradient descent of the loss function with the objective of finding an initial model (i.e., meta-model) for each client.

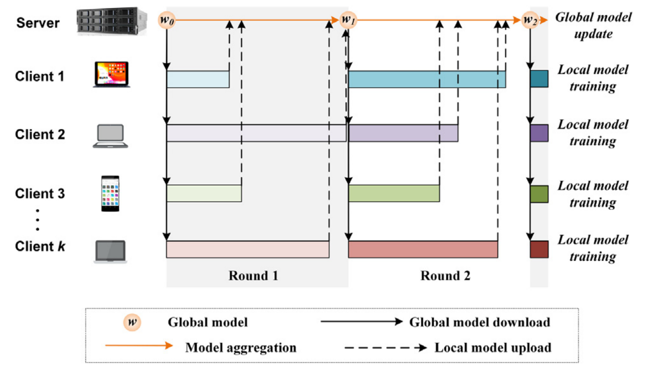


Fig. 7. The synchronous aggregation in FL. The diagram depicts the synchronous aggregation process, where the server carries out model aggregation in each round after receiving updates from all participating clients.

4.3. Taxonomy of model aggregation

As FL research advances, the possibilities for designing model aggregation methods seem almost limitless. There are various aggregation techniques that serve different purposes, such as enhancing the performance of the federated model, minimizing communication overhead, and addressing data security and privacy concerns. This paper categorizes the aggregation techniques into four types based on their aggregation form: synchronous, asynchronous, hierarchical, and robust aggregation.

4.3.1. Synchronous aggregation

In synchronous aggregation-based FL research, model aggregation occurs after all client updates have reached the server, and the latency or lag experienced on the client side is generally disregarded. Fig. 7 illustrates the schematic diagram for synchronous aggregation. In algorithms like FedAvg, for example, the server randomly selects clients to participate in training during each round of communication, and their model parameters are included in the aggregation process. The synchronization occurs without explicitly considering the latency or potential delays experienced by individual clients.

Even though the server performs synchronous aggregation of client updates, discrepancies can still arise during the aggregation process, giving synchronous-based FL research its unique advantages. To tackle data heterogeneity in the synchronous scenario, recent research works often focus on (1) Adding regularization terms to constrain local models; (2) Adjusting aggregation weights to assign appropriate shares for different clients; (3) Compressing the model to improve its expressiveness.

- **Regularization terms introduction.**
In the case of non-IIDness, the optimization objective of the local model differs from that of the global model, which can lead to increased convergence time for the global model or even cause it to diverge. For example, in the FedDyn approach [54], a dynamic regularizer is suggested for each client in each round to minimize the disparities between local models and the global model. This is due to the fundamental difference between the minimum value of the local-device empirical loss and that of the global empirical loss. In the FL system called FedDist [55], regularization terms are based on Euclidean distances between the local models and the global model. On the other hand, in the FedAAR approach introduced by [56], cosine distances are employed as a measure of dissimilarity between the local models and the global model.

- **Weights adjustment.**
When aggregating models using the parameter-weighted average method, the typical approach is to determine the aggregation weights based on the amount of data each client possesses. However, this approach can be disadvantageous for data-poor clients. Clients with larger amounts of data have a greater influence on the quality of the global model, while those with smaller amounts of data may not receive sufficient attention [57,58]. To address this issue, researchers have proposed optimizing the allocation of aggregation weights [59–61]. The aim is to ensure a fair distribution of weight by assigning appropriate shares to different clients. This approach considers clients that are often overlooked, allowing global models to learn additional knowledge from these previously neglected clients [61]. For instance, in the FedDisco approach introduced by [62], the authors found that using dataset size as the sole aggregation weight is suboptimal. Instead, they propose using the difference between local and global category distributions as a complementary metric for aggregation weights. Similarly, in the work by [63], the authors suggest adaptively assigning different weights to clients based on their contribution in each round. In their work, Wu et al. [64] propose measuring the contribution of participating clients by contrasting the local gradient vector with the global gradient vector. They suggest quantizing the weights using a nonlinear mapping function. Another efficient approach for quantifying weights is based on a scoring system. In studies such as [38,65–67], reputation scores are introduced for local models and utilized to scale the aggregation weights. The reputation score of a client is computed based on performance metrics of the local model in each round of training, providing an assessment of the direct contribution of the local model [63,64]. Furthermore, the learning mechanism, such as an attention mechanism, can be employed to learn learnable parameters in the model for obtaining aggregated weights [68,69]. Some researchers [70,71] propose incorporating an attention mechanism during model aggregation, where an attention score is learned for each client, and these attention scores serve as the weight factors. In the FedLAW framework [72], a learnable approach to aggregation weights is proposed. The authors demonstrate that the L1 norm of the aggregation weights can be less than 1, indicating that the aggregation weights can be adaptively adjusted during the aggregation process.
- **Knowledge distillation.**
Knowledge distillation, commonly used in ML, involves training a smaller model (student) to learn from a larger model (teacher). In the context of FL, applying knowledge distillation techniques is referred to as federated distillation [73,74]. This technique has the potential to significantly reduce communication costs in FL [75,76]. The general framework of knowledge distillation in FL is illustrated in Fig. 8. For instance, the FedGKT framework proposed by [77] trains smaller networks on edge nodes and employs knowledge distillation to transmit the learned knowledge to the server. This approach helps reduce communication costs, particularly when the local model is a large convolutional neural network. Knowledge distillation techniques have been utilized in various FL architectures to enhance convergence speed and performance. Some studies, such as [78,79], leverage knowledge generated on public datasets to facilitate convergence. Distillation terms are also added to the local objective function in order to generate personalized models, as demonstrated in [80,81]. Furthermore, integrating local knowledge with predictive logits can improve the performance of general distillation fusion models, as observed in [82].

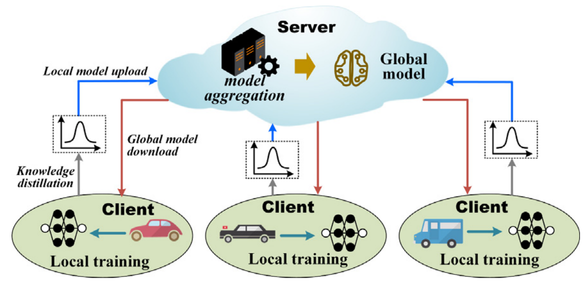


Fig. 8. Knowledge distillation in FL.

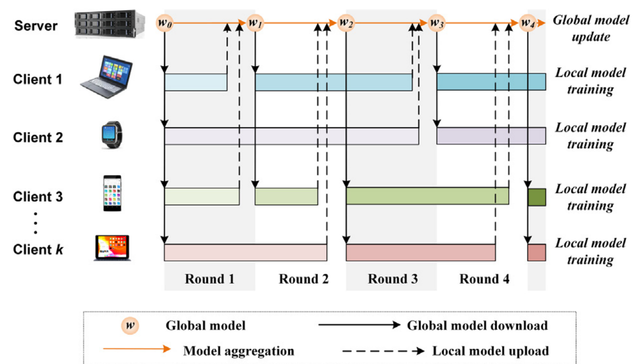


Fig. 9. Asynchronous aggregation in FL. In the asynchronous aggregation process, the server carries out the aggregation in a set manner, as soon as updates from individual clients are received, without requiring synchronization with all clients.

4.3.2. Asynchronous aggregation

Asynchronous FL (AFL) has gained significant attention due to the growing influence of device heterogeneity in federated systems. AFL allows clients to upload their local updates in a staggered manner, which helps mitigate the negative impact of device heterogeneity. In traditional FL, delays caused by poor network signals or client crashes during training may lead to delays in uploading updates, thereby increasing the waiting time for the server to receive updates from all clients [83,84]. The primary goal of asynchronous aggregation is to accelerate the training process. In fully asynchronous aggregation, the aggregation takes place as soon as the server receives the local updates from each client, allowing each client to train independently without waiting for other clients to complete their updates. This asynchronous approach enhances the efficiency and scalability of FL by reducing the waiting time and enabling clients to contribute their updates at their own pace. A schematic diagram of asynchronous aggregation is shown in Fig. 9. Recent research has proposed further improvements, such as (i) dynamic fusion and (ii) evaluation of the client-side model, optionally for aggregation. In addition, the semi-asynchronous aggregation between synchronous and asynchronous also brings about the optimization of the training process.

- **Dynamic fusion of local models.**
Dynamic fusion of local models has emerged as a popular method in FL to reduce overall training time [85–87]. While fully asynchronous aggregation performs model aggregation as updates arrive [88], this approach can result in low-accuracy global models in each round, necessitating multiple rounds of training to achieve satisfactory accuracy. To overcome this challenge, researchers have proposed dynamic fusion techniques based on interval time windows or the number of clients [89,90]. For instance, FedPA [91]

addresses the trade-off between training time and model performance by dynamically determining the number of aggregations in each round. Another approach involves handling stale models in the aggregation process [92], where local models with staleness exceeding a threshold are excluded from the aggregation. In terms of adaptive deadline determination, methods such as [44,93] have been proposed. These methods calculate the expected time for each round based on the computing resources and channel conditions of each mobile device, enabling the adaptation of deadlines to improve the overall FL process. In addition to the aforementioned client selection and dynamic aggregation methods, several other strategies have been proposed to enhance asynchronous aggregation. These strategies include excluding clients who have not completed their tasks in the current aggregation round and evaluating the decision to aggregate based on a client's historical performance and current status [94–97]. In [98], the uplink and downlink processes are planned to ensure a balanced influence and share among all clients. Furthermore, some researchers [99,100] have proposed employing temporal weight decay strategies to achieve effective asynchronous aggregation.

- **Clients evaluation.**
  The evaluation of client performance plays a significant role in the domain of AFL [101,102]. Due to varying signal strengths and limited network resources, not all clients can effectively participate in the aggregation process. Moreover, the statistical heterogeneity of local models can negatively impact the global model. Therefore, prioritizing clients with better communication capabilities, higher prediction accuracy, or models more aligned with the optimal global model is crucial [103]. For instance, in the Eiffel system [104], a mobile edge computing framework, clients are selected for aggregation based on relevant metrics such as data size, computing power, and last update time. An overall index is calculated to determine the priority of each client. Other methods for client selection include sorting clients based on the gradient update norm [105], utilizing radial-basis-function [106], assessing model uncertainty [107], and considering the Mutual Information ratio between ground truth and model predictions [108]. Researchers have also proposed bandit learning algorithms and Lyapunov optimization techniques to address the asynchronous client selection problem [109]. Some studies employ the Sequential Kalman Filter to rank the parameters uploaded by clients [110,111]. Additionally, the quality of clients can be evaluated using various approaches, as demonstrated in works such as [112,113]. These evaluation methods contribute to the selection of suitable clients for aggregation, enhancing the performance and effectiveness of asynchronous aggregation in FL.
- **Semi - asynchronous.**
  Semi-asynchronous aggregation serves as an intermediary aggregation method between synchronous and asynchronous aggregation [114]. In synchronous aggregation, the server must wait for all resulting client models to arrive before performing aggregation, leading to potential long waiting times. On the other hand, fully asynchronous aggregation addresses issues like device heterogeneity but may result in frequent model transfers, consuming significant communication resources. To address these challenges, researchers have proposed semi-asynchronous FL mechanisms such as FedSA [115]. With a predefined communication budget, FedSA allows the server to perform a certain number of aggregations based on the order of client model arrivals in each round, thus optimizing the trade-off between waiting time and resource consumption. Similarly, SAFA [116] is a semi-asynchronous FL protocol that introduces the hyperparameter "lag tolerance" to coordinate lagging clients and compensate for the arrival order of client models. Additionally, [117] proposed a semi-asynchronous aggregation strategy that fixes client selection in the first round and subsequently selects clients based on priority. Semi-asynchronous aggregation strikes a balance between synchronous and asynchronous aggregation, effectively improving the round efficiency of the FL process while minimizing resource waste. It offers a practical compromise to leverage the benefits of both synchronous and asynchronous approaches in FL.

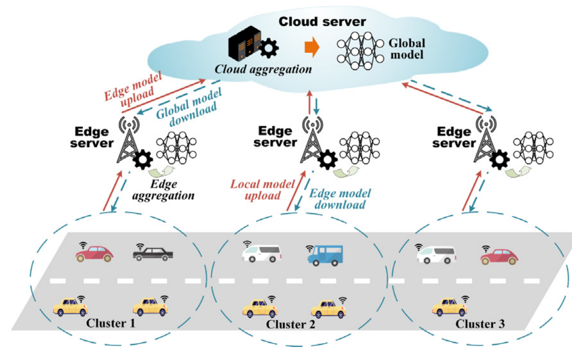


Fig. 10. Hierarchical aggregation in FL. The figure depicts the hierarchical aggregation process through the federated IoV: vehicles are clustered into different clusters, and upload local models to the corresponding edge servers (i.e. roadside units), which then upload the models to the cloud for further global model updates.

4.3.3. Hierarchical aggregation

In the context of edge learning, the presence of a large number of edge devices, such as IoT devices, can significantly impact the efficiency of FL due to frequent model aggregation and the resulting high communication overhead [118]. To tackle this challenge, researchers have proposed a hierarchical aggregation approach that incorporates an edge layer to partially aggregate local models from closely related client devices before further aggregation on the cloud server [119,120]. This hierarchical approach aims to reduce communication overhead and the number of model transfer rounds [121] by introducing multiple aggregation centers. Fig. 10 illustrates the architecture of hierarchical FL, where the hierarchy optimization and client similarity clustering are two key research focuses.

- **Hierarchical optimization.**
  Improving the hierarchical structure of Hierarchical Federated Learning (HFL) can effectively address resource allocation and communication efficiency challenges [12,122–124]. Various approaches have been proposed to enhance the hierarchical architecture of HFL and optimize its performance. One example is the FogL architecture proposed in [125], which adopts a tree-based federated network structure. In this architecture, end devices are represented as leaves, and the master server acts as the root. By employing multi-level and multi-stage tree aggregation techniques [126,127], FogL mitigates the risk of system overload and improves resource allocation efficiency. In the context of the Internet of Vehicles (IoV), HFL research [128–130] has explored the utilization of roadside units as the middle layer and the road traffic cloud as the cloud aggregator. This hierarchical approach leverages the infrastructure of the IoV to facilitate communication and coordination among vehicles. However,

it is worth noting that traditional HFL approaches may encounter performance degradation for highly mobile users in hierarchical wireless networks, as discussed in [131]. To address this challenge, the Mobile-Aware Collaborative Federated Learning (MACFL) algorithm is proposed. MACFL allows mobile users to download the cluster model from the nearest edge server for local training, and if they roam to another cell, they can upload updates to a different edge server, thus improving communication efficiency. Incentive mechanisms can also be introduced to enhance communication efficiency in HFL. For instance, in [132], a data owner competition framework is proposed where participants compete for cluster head qualification, equivalent to edge aggregators. Rewards are distributed based on their contributions, incentivizing active participation and efficient communication. Another approach is MaxQ [133], which applies game theory to improve matching mechanisms in HFL, enabling efficient collaboration and resource allocation.

- **Clients clustering.**
The idea of clustering in Hierarchical Federated Learning (HFL) has gained significant attention in recent research [134,135], as it offers a promising approach for improving system performance and resource utilization. Clustering involves grouping clients with similar characteristics or data distributions into clusters, and performing model aggregation within each cluster.
In this context, several clustering-based approaches have been proposed in the literature [136–138]. For example, Lin et al. [139] divide clients into clusters based on their communication capabilities and leverage device-to-device (D2D) communication within each cluster. Periodic global aggregations are then performed to update the global model. The FedSim framework [140] utilizes the k-means algorithm to measure similarity between clients and guide the aggregation process. Different distance metrics, such as L1 (Manhattan distance), L2 (Euclidean distance), and cosine distance, have been employed to quantify the similarity between clients in clustering-based approaches [141–143]. Another line of research focuses on clustering clients based on the distribution of their local data [144,145]. For instance, the HPFL-CN framework [146] proposes aggregating edge devices with similar environmental data distributions and efficiently training personalized models for each cluster using a hierarchical architecture. In the work by Bao et al. [147], clients are clustered into non-overlapping coalitions based on the distance between their data distributions and the volume of their data. Each client collaborates only with clients that have similar data distributions, promoting effective model training. Clients with less data tend to collaborate with a larger number of other clients to compensate for their limited data.

4.3.4. Robust aggregation

Compared to centralized ML, FL provides greater data privacy and security guarantees. However, recent research has identified several security vulnerabilities in FL [148,149]. For example, inference attacks can be launched by an attacker to infer the data distribution of participants by analyzing the parameters of local models [150], while backdoor attacks can be initiated by malicious actors by introducing bad clients to the global model [21]. These safety risks have spurred researchers to devise strategies to enhance the robustness of FL models [151]. To ensure secure aggregation throughout the FL process, researchers recommend the use of various encryption techniques, such as differential privacy [152] and homomorphic encryption [153]. Differential privacy involves adding random noise to the output to prevent attackers from reverse-engineering sensitive data, while homomorphic encryption enables basic functional operations to be performed on encrypted data. In addition, the use of decentralized model aggregation is suggested, for example, by leveraging blockchain technology [154] or gossip principles [155]. This decentralized approach eliminates the central server from the system, thereby preventing single points of failure. Other approaches include optimizing contract mechanisms [156], using robust stochastic model aggregation [157], electing a small committee, rather than assigning sanitization factors [150] or avoiding small group domination. Furthermore, strategies like multi-party computation (MPC), and trusted execution environment (TEE) demonstrate the ongoing effort to address security vulnerabilities in FL and enhance the privacy protection of the system.

- **Homomorphic encryption-based.**
In homomorphic encryption (HE) based schemas, the users encrypt their models using the same public key, which allows the central server to add the encrypted models together using the additive homomorphic property of the underlying crypto-system. The ownership of the secret key is a crucial point in public-key crypto-systems used for FL aggregation. Following the taxonomy in [158], there are three settings of secret key management. In the first setting, the secret key is shared among all users but kept confidential against the central server, making the global model public to all participants. Many different crypto-systems have been adopted in this setting: RSA [159], Paillier [160], lattice-based [161], BGN [162], ElGamal [163]. In the Second setting, the secret key is known only by the central server to protect the privacy of the global model [164]. The drawback of this approach is that the server can decrypt the encrypted models sent by users, thus breaking their model privacy. Other privacy-preserving methodologies are required to improve privacy guarantees, such as masking models [165] or the use of a trusted party to manage the secret key [166,167]. The third one can be seen as a way to improve the security of the first setting, where a set of users share the same secret key. More than a threshold number of users must cooperate to decrypt an encrypted message. Crypto-systems such as threshold Paillier [168,169], ElGamal [170] are used in this setting. This setting provides a higher security guarantee than the first setting, as the central server must collude with more users to break the security.
- **MPC-based.**
Several works have explored the use of Multi-Party Computation (MPC) to enable privacy-preserving aggregation and FL training [171,172]. In these schemes, users distribute their locally trained models to a set of selected users/servers referred to as agents, who subsequently aggregate them to construct a new global model. In [173], the authors propose the utilization of a Fast Fourier Transform (FFT) based secret sharing scheme instead of the traditional Shamir secret sharing scheme. Alternatively, verifiable sharing schemes have been utilized in [174,175]. Furthermore, [176,177] propose an aggregation strategy based on a two-step process where users first elect a committee, which then receives and aggregates the models shared by the users. Other methodologies rely on sharing the models between two [178,179] or more [174,180] servers to facilitate privacy-preserving aggregation and FL training.
- **Blockchain-based.**
The utilization of blockchain technology [181], which is a decentralized and non-tamperable distributed ledger, offers

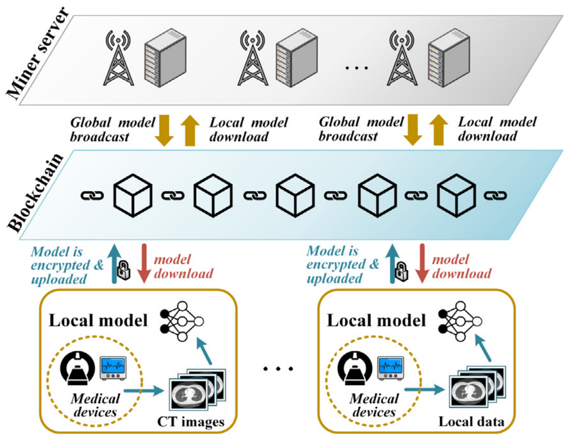


Fig. 11. An example of blockchain-based FL.

a wide range of advantages for various research fields such as healthcare, finance, and education [182]. In recent years, more researchers have been exploring the combination of FL with blockchain technology [154,183,184]. In Fig. 11, a blockchain-based architecture for FL is illustrated where a client (e.g., medical devices) first downloads the current global model from the blockchain for initialization. Subsequently, the client encrypts and uploads the trained model to the blockchain, which is then downloaded by the miner server. The server aggregates the models and broadcasts the new global model to other nodes. In a recent study [185], each aggregation node conducts a quality test on the local model and broadcasts the reputation evaluation to the blockchain network. The reward distribution is based on the combination of client contributions and reputation. This reputation-based reward distribution algorithm, along with blockchain technology, provides quality assurance for the model [186,187].

- **TEE-based.**
A Trusted Execution Environment (TEE) is a secure area in the main processor that enables the storage, processing, and protection of sensitive data and code within a trusted and isolated environment [188]. TEEs represent a secure counterpart to the Rich Execution Environment (REE), which is the standard operating system that the device is running. Within FL frameworks, TEEs can be successfully used for secure aggregation: the users encrypt their locally trained models and transfer them to the REE. The TEE then receives the encrypted models from the REE, decrypts them, and aggregates them. The output is returned to the REE for distribution to all users [189,190]. Several strategies have been proposed to enhance the security and privacy of FL systems utilizing TEEs. These include distributing trust among several TEEs [191], applying DP techniques to perturb users' models before uploading them to the TEE [190], and utilizing a TEE to add randomness by shuffling users' models before uploading them to the central server. Furthermore, a novel approach for secure aggregation in FL using TEEs is to deploy ML training algorithms inside TEEs themselves [192,193]. This approach provides an additional layer of security by ensuring the confidentiality of data and code.

4.4. The proposed taxonomy

To provide a comprehensive overview of the emerging research methods in FL, we have classified them into four categories and provided detailed explanations. The taxonomy is depicted in Fig. 12, and further information can be found in Table 3, which presents the specific details of the taxonomy along with corresponding literature references. In general, synchronous aggregation is effective in achieving good model performance when the number of devices in the federated system is relatively small. However, as the number of clients increases, client heterogeneity becomes more pronounced, leading to longer training times as clients wait for each other. In such scenarios, asynchronous aggregation becomes a more favorable choice. Hierarchical aggregation, on the other hand, is particularly suitable for large-scale IoT applications like the Internet of Vehicles (IoV), where a significant number of devices require coordination. This approach helps to reduce communication overhead and the number of model transfer rounds. For applications with high system security requirements, the robust aggregation method is preferred. This approach addresses security concerns by incorporating techniques such as parameter encryption or blockchain technology, which enhance the overall security of the federated system. By categorizing the research methods and considering their specific characteristics and benefits, FL practitioners can make informed decisions on selecting the most appropriate model aggregation approach for their specific requirements and constraints.

5. Applications of aggregation methods

FL's unique advantages have made it an attractive solution for a variety of fields, such as smart transportation, financial business, and healthcare [198]. FL can handle data islands while providing strong privacy protection. As FL gains more attention in real-life applications, including smart healthcare, smart transportation, smart city, smart industry, etc. In practical applications, selecting the appropriate aggregation method is crucial since it has a significant impact on the functionality and efficiency of the designed FL architecture. In the following discussion, we will explore the reasons and benefits of choosing different aggregation methods for various industries from the perspective of practical application.

5.1. Model aggregation in smart healthcare

In the healthcare industry, ensuring user privacy is of paramount importance. While the rapid development of ML has greatly promoted the advancement of smart healthcare, the leakage of medical data has also caused significant problems for people. To address this issue, multiple means of enhancing privacy protection can be employed in FL based on robust aggregation, thereby helping to establish a safe and secure smart medical system [199–202]. For example, Kumar et al. propose the *MediSecFed* system [203], a security framework for FL in hostile environments. The authors demonstrate that the *MediSecFed* system is also robust to poisoning attacks through experimental evaluations on real-world pneumonia datasets. In [10], the authors propose an FL framework for protecting medical privacy using blockchain technology and homomorphic encryption methods. Evaluation experiments on medical images from CT scanners demonstrate that the framework can strike a balance between privacy and accuracy. In addition, the synchronization-based aggregation scheme helps to improve the prediction accuracy of the FL model. Choudhury et al. [204] conducted a study of electronic health data to predict adverse drug reactions. Their experimental results show that the FL model performs similarly to centralized ML and avoids the challenges associated with the latter.

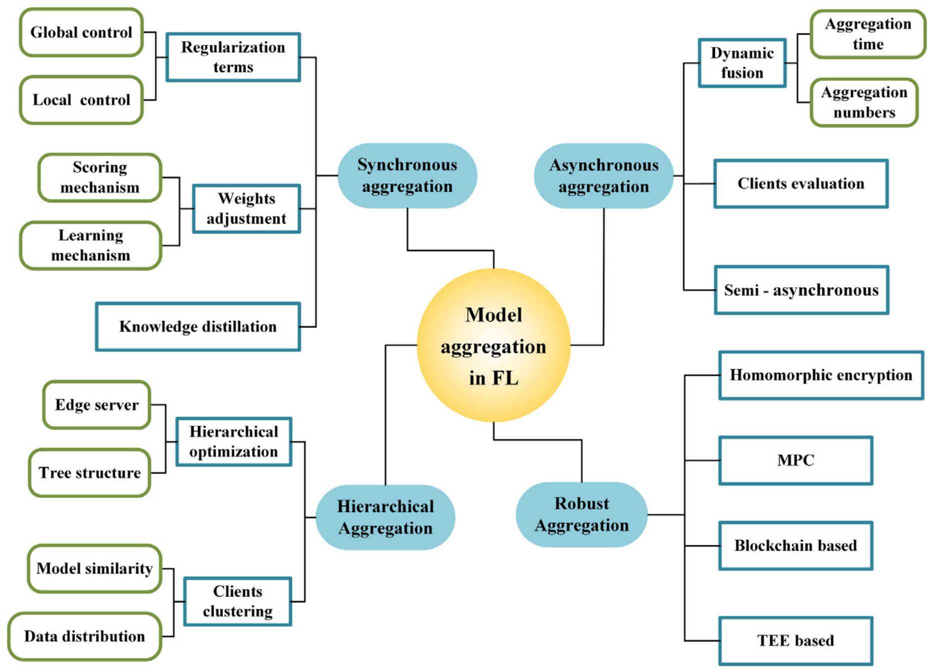


Fig. 12. A diagram of the proposed taxonomy of model aggregation techniques in FL.

Table 3
Details of the proposed taxonomy and the corresponding literature.

Synchronous aggregation	Regularization terms		Weights adjustment		Knowledge distillation	
	Global control	Local control	Scoring mechanism	Learning mechanism		
	[34,194]	[54–56]	[38,59,60], [63–65] [66,67]	[68,70,71] [69,72]	[73–82,195]	
Asynchronous aggregation	Dynamic fusion		Clients evaluation		Semi - asynchronous	
	Aggregation time	Aggregation numbers				
	[44,83,86,87] [84,88–90] [92,93,99,100]	[85,91,94] [95,96]	[101–113]		[114–117,196,197]	
Hierarchical Aggregation	Hierarchical optimization			Clients clustering		
	Edge server		Tree structure	Model similarity		Data distribution
	[118–121,128–130]		[125–127]	[134–143]		[144–147]
Robust Aggregation	Homomorphic encryption		MPC based	Blockchain based		TEE based
	[153,158–170]		[172,173,176] [174,175,177] [178–180]	[154,156,183–187]		[189–191] [192,193]

5.2. Model aggregation in smart transportation

ML technology has been commonly used to improve transportation, and recently, FL has been gaining attention in this field. By surveying the relevant literature, we found that several different aggregation methods are widely adopted by researchers in the field of intelligent transportation, thanks to the different benefits provided by each aggregation method [205]. In FL studies based on the IoV, hierarchical and asynchronous aggregation schemes are commonly used due to the large number of vehicles involved and the need for full coordination [206]. Furthermore, roadside units in IoV can provide intermediate support for staged aggregation [207]. [208] is a FL-IoV framework based on hierarchical aggregation, which realizes dynamic map fusion technology without data labels. In this framework, roadside units provide labels for local training, and cloud servers perform model aggregation. [209], vehicle edge computing was studied using FL, and a client selection approach was used to improve the accuracy and efficiency of model aggregation. Specifically, by evaluating the local image quality and computing power, a good local DNN model is selected and sent to the central server. In [210], an asynchronous aggregation method is used for local model aggregation of vehicles in IoV. Because in each round of communication, a large number of vehicles upload model parameters to the server in the uplink communication phase, which will cause huge communication pressure and prolong the training time, they based their framework on the rules of partial vehicle participation, achieving faster convergence in fewer communication rounds.

5.3. Model aggregation in smart city

FL can be used in smart cities to dynamically control environmental pollution, which can help protect the lives and health of urban residents [211]. FL can be used to train machine learning models on data collected from sensors, such as air quality sensors, distributed throughout a city. By aggregating these models, FL can provide insights into environmental pollution levels and help improve decision-making on how to reduce pollution and protect public health. Extensive research has been conducted on the application of FL in smart cities, and asynchronous aggregation has been found to be a popular method for coordinating the aggregation of client models in FL. In FL based on asynchronous aggregation, the aggregation of client models can be dynamically coordinated to improve the operating efficiency of the federated system. The study by Gao et al. [85] proposes an asynchronous FL model aggregation method called *n-softsync* for air quality prediction, which limits the number of local models when aggregation to reduce time overhead. In the model aggregation step, only *n* nodes are allowed to upload the latest local model in each round; the remaining nodes that have not uploaded their local models use their old models for the aggregation. Another study [88] proposes an asynchronous FL framework for urban environment perception that considers regional characteristics of monitoring points and uses an asynchronous aggregation strategy to address different bandwidth requirements and to reduce latency. [146] proposes an edge-intermediary-cloud FL architecture to achieve prediction tasks in urban environments. In this framework, environmental data are mapped to different complex network domains. Similarity clustering is performed by extracting low-level feature representations of each edge server. The intermediary is responsible for coordinating the model training of the edge servers in each cluster, and periodically uploading the weights to the cloud server for model aggregation.

5.4. Model aggregation in smart industry

In the field of the Industrial Internet of Things (IIoT), research related to FL pays more attention to the robustness of the system. This is because IIoT involves a large number of important industrial devices, and if the nodes of the federal network are attacked or damaged, the industrial system may be paralyzed. Thus, FL based on robust aggregation is more commonly used in IIoT research [176]. Traditional FL can have a single point of failure when the central server is damaged. To address this issue, researchers have proposed different methods for secure and decentralized FL. For instance, a serverless FL framework driven by blockchain was proposed in [212] for distributed ML in smart grid analysis. The authors used asymmetric cryptography to encrypt models and verified the effectiveness of the proposed method using power grid simulation datasets. In [184], a framework based on blockchain and with Intel Software Guard Extensions (SGX) is proposed for simulating the scenario of FL in smart warehouses. Each of these blockchain nodes hosts a trusted SGX processor for secure aggregation of models. In this framework, each blockchain node can verify the authenticity of the aggregated model, and the blockchain consensus mechanism is run to ensure the integrity of the model.

5.5. Model aggregation in other fields

FL has found applications in other domains, such as finance, education, network security, E-commerce recommender system [213], natural disaster prediction [214] etc. The efficient model aggregation has brought great advances to those industries. For example, Imteaj et al. [215] utilize asynchronous FL to predict customers' financial distress. In this framework, local models learn about customers' personal information and past financial status, and accurate predictions are achieved by considering clients' local models at different epochs. Furthermore, with the continuous development of network technology and the popularization of network applications, research on network security issues based on FL has become a research hotspot. Zhao et al. [216] proposed an intrusion detection system called *DAFL*. In addition to the inherent data protection advantages of FL, the dynamic filtering and weighting strategy achieves less communication overhead and higher detection performance. In recent years, online course systems have gradually become popular in the education industry and schools, and data leakage risks have also received a lot of attention. An integrated system of multimedia course recommendation and data-secure FL is proposed in the study [217], which keeps a copy of the recommended model on each user device and coordinates the division of model aggregation communication rounds. Experiments on datasets containing real multimedia courses demonstrate effective operation in a privacy-preserving mode.

As a conclusion, since different application domains have different characteristics and situations, there are generally one or two main aggregation methods adapted to meet application requirements. Table 4 summarizes the main model aggregation methods in different domains.

6. Challenge and future directions

FL has led to the development of several aggregation methods, but there are still some significant challenges that need to be addressed. In this section, we will discuss these challenges in detail, focusing on three aspects: statistical heterogeneity, communication bottlenecks, and secure aggregation. We will also highlight potential future trends in addressing these challenges.

6.1. Statistical heterogeneity

In FL, a significant challenge in model aggregation is caused by statistical heterogeneity [218]. The FedAvg algorithm performs model aggregation by taking a weighted average of local model parameters from all participating parties. However, this may cause the aggregated global model to converge poorly or even diverge. To address this issue, some studies have employed the Bayesian non-parametric mechanism to perform model aggregation through neuron matching and a combination of local models, as seen in PFNM [219] and Claici et al. [220]. In addition, Shukla et al. [221] proposed the *Infogain FedMA* algorithm, which uses an information-gain-based sampling method to select the parameters to be aggregated, paired with probabilistic federated neural matching. However, these methods have only been applied to relatively simple neural network models, and generalizing them to complex network models to improve applicability remains a challenge that needs to be further explored.

6.2. Communication bottlenecks

Model aggregation in FL faces another significant challenge, namely the communication problem [222]. This problem directly affects the aggregation speed, which slows down the overall training progress. In a typical federated system, there are usually multiple clients, such as hundreds or thousands of devices in an IoT-based federated system. During model aggregation, a large number of clients need to upload their local updates to the same network, which can cause severe communication congestion problems due to limited network bandwidth [223]. This is also a major challenge faced by FL in enterprise implementation. While training simple models in FL can reduce the communication burden, the explosive growth of data in the internet age means that this may not be sufficient to meet practical application needs. Therefore, training large network models has become

Table 4
A summary of the main aggregation methods per each application domain.

Applied domain	Aggregation	Paper	Method	Purpose
Smart Healthcare	Robust Aggregation	[202]	Use differential privacy and cryptography.	Increase security level during model aggregation; Protect users' medical privacy.
		[203]	Exchange and encode model logits instead of parameters.	Protect the system from poisoning attacks; Protect users' medical privacy.
		[10]	Utilize blockchain technology and homomorphic encryption.	Decentralize and secure update gradients; Protect medical privacy.
	Synchronous aggregation	[204]	Assign higher weights to clients with rare events.	Achieve model performance comparable to centralized ML; improve medical disease diagnosis accuracy while preserving medical data.
Smart Transportation	Hierarchical Aggregation	[207]	Roadside units as intermediate aggregators and cloud servers as final aggregators.	Coordinate a large number of clients (vehicles); Ensure the orderly operation of the Federal IoV.
		[208]	Roadside units provide labels for local training, and cloud servers perform model aggregation.	Coordinate a large number of clients (vehicles); Ensure the orderly operation of the Federal IoV.
	Asynchronous aggregation	[209]	Good local models are selected for aggregation by evaluating clients.	To improve the efficiency of model aggregation; Reduce training time.
		[210]	Create rules for partial vehicle participation.	To reduce the pressure of uplink and downlink communication during model aggregation.
		[206]	Evaluate local model versions for aggregation.	To reduce model training time and bandwidth costs.
Smart City	Asynchronous aggregation	[88]	Model aggregation is performed as soon as the local model is uploaded.	Reduce invalid latency and bandwidth requirements.
		[85]	Only n nodes are allowed to upload the latest local model in each round.	Reduce time overhead.
		[211]	The server first aggregates models related to regions.	Reduce time overhead; Improve the efficiency of model aggregation.
	Hierarchical Aggregation	[146]	Clients are clustered based on similarity, and edge servers and cloud servers perform aggregation in turn.	Reduce the communication overhead of the FL system.
Smart Industry	Robust Aggregation	[212]	Asymmetric cryptography for encryption and smart contracts in the blockchain to drive training.	Prevent single point of failure and improve system security.
		[176]	Two-stage secure multi-party computation.	Improve system security.
		[184]	Relying on blockchain and processors with Intel Software Guard Extensions.	Secure aggregation to prevent model tampering.

a focus of several studies [125]. Correspondingly, maximizing the use of limited resources and improving aggregation efficiency under such conditions is an urgent problem that needs to be solved.

There are proposed solutions to address the communication bottleneck in FL. One such solution is FL based on over-the-air computation (AirComp), which can achieve fast model aggregation. AirComp is a non-orthogonal multiple access (NOMA) technique that uses the waveform superposition characteristics of multiple access channels to perform combined calculation of data transmitted by multiple clients. Ni [224] et al. proposed to use intelligent reflective surfaces (IRS), which is a technique to reconfigure the wireless propagation environment, to improve signal distortion caused by AirComp. Another proposed solution is to deploy multiple relays to assist signal transmission and improve the performance of aerial model aggregation, as suggested by Lin et al. [225]. Additionally, the sixth generation (6G) wireless communication is expected to be an effective method to address the communication bottleneck in FL. Compared with previous generations of wireless communications (4G and 5G), 6G has higher data transmission rates, wider frequency bands, and wider network coverage [226]. The 6G era is expected to help FL address its model aggregation difficulties, and in turn, FL can promote the integration of 6G into more IoT industries and AI services.

6.3. *Secure aggregation*

In FL, security is a critical issue that needs attention, especially in model aggregation. In centralized FL, the server is assumed to be honest but curious, and the clients are honest. However, the presence of attackers can easily disrupt this assumption [227]. In FL, the attacker can take active and passive attacks [21]. Active attacks include model poisoning attacks [20]. Generally speaking, in a model poisoning attack, the attacker can modify the local model before uploading it to the central server, thereby affecting the aggregated global model. The idea of filtering out suspect local models for evaluating client-side updates has emerged in several studies [156]. When the vast majority of clients are honest, filter-based approaches can lead to good aggregated results. But when there is more than one dangerous client, there may be a Sybil attack by multiple attackers which will cause greater damage to the model. At this time, the filtering-based method may also cause the loss of client information, which is a method that is not worth the candle. How to ensure safe aggregation without losing information is a matter of balance. On the other hand, passive attack means that the attacker does not change the training process of FL, but makes inferences by observing updates, such as the inference attack. In an inference-based attack, the model parameters are reversely deduced, which will lead to the leakage of private data information. Both participating clients and malicious centralized servers in FL have the potential to launch inference attacks.

Currently, researchers are exploring the potential benefits of blockchain technology, such as decentralization, traceability, and irreversibility, in addressing security concerns in model aggregation [228]. Additionally, a trusted environment is necessary for model aggregation [229]. In addition to these technical solutions, proper governance and regulation are also essential to ensure the security of FL systems. This includes establishing a clear and transparent data collection and usage policy, ensuring that data is collected and processed lawfully, and obtaining the necessary consent from individuals. It is also important to implement effective monitoring and auditing mechanisms to detect and respond to security breaches. Finally, it is essential to educate both clients and developers about the security risks associated with FL and how to mitigate them.

7. Conclusion

In this paper, we have conducted a comprehensive survey and discussion on the topic of model aggregation in FL. FL has emerged as a distributed machine learning paradigm with enhanced privacy protection, and model aggregation plays a crucial role in this context. Our motivation for this paper stems from the lack of a comprehensive survey and taxonomy of model aggregation in FL.

We first introduced our research methodology and provided an overview of FL and model aggregation. We categorized model aggregation into four forms: synchronous, asynchronous, hierarchical, and robust. We then explored the benefits of model aggregation in practical applications, such as smart healthcare and smart transportation.

Additionally, we discussed the current challenges in model aggregation in FL and proposed potential future research directions. Our survey is the first to provide a taxonomy of the different forms of model aggregation in FL, serving as a foundation for further exploration and development in this area. Through our extensive discussions, we aim to foster more research interest in the field of FL.

CRediT authorship contribution statement

Pian Qi: Data Curation, Investigation, Methodology. **Diletta Chiaro:** Investigation, Conceptualization, Methodology, Formal analysis. **Antonella Guzzo:** Conceptualization, Writing - Review & Editing. **Michele Ianni:** Writing - Review & Editing, Investigation, Methodology. **Giancarlo Fortino:** Writing - Review & Editing, Validation. **Francesco Piccialli:** Conceptualization, Methodology, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

This work was partially supported by the following projects:

- PNRR project FAIR - Future AI Research (PE00000013), Spoke 3, under the NRRP MUR program funded by the NextGenerationEU.
- PNRR Centro Nazionale HPC, Big Data e Quantum Computing, (CN_00000013)(CUP: E63C22000980007), under the NRRP MUR program funded by the NextGenerationEU.
- PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.
- POS RADIOAMICA project funded by the Italian Ministry of Health (CUP: H53C22000650006)

References

[1] A. Paleyes, R.-G. Urma, N.D. Lawrence, Challenges in deploying machine learning: a survey of case studies, ACM Comput. Surv. (2020).
[2] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[3] J. Konečnỳ, H.B. McMahan, F.X. Yu, P. Richtárik, et al., Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint.
[4] C. Pappas, D. Chatzopoulos, S. Lalis, M. Vavalis, Ipls: A framework for decentralized federated learning, in: IFIP Networking Conference, IEEE, 2021, pp. 1–6.
[5] S. Ji, J. Zhang, Y. Zhang, Z. Han, C. Ma, LAFED: A lightweight authentication mechanism for blockchain-enabled federated learning system, Future Gener. Comput. Syst. 145 (2023) 56–67.
[6] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.
[7] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, et al., Federated optimization in heterogeneous networks, in: Proceedings of Machine Learning and Systems, Vol. 2, 2020, pp. 429–450.
[8] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Int. Things J. 3 (5) (2016) 637–646.
[9] W.Y.B. Lim, N.C. Luong, D.T. Hoang, Y. Jiao, et al., Federated learning in mobile edge networks: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2031–2063.
[10] R. Kumar, J. Kumar, A.A. Khan, H. Ali, et al., Blockchain and homomorphic encryption based privacy-preserving model aggregation for medical images, Comput. Med. Imaging Graph. 102 (2022) 102139.
[11] S. Wang, F. Liu, H. Xia, Content-based vehicle selection and resource allocation for federated learning in iov, in: Wireless Communications and Networking Conference Workshops, IEEE, 2021, pp. 1–7.
[12] X. Chen, G. Zhu, Y. Deng, Y. Fang, Federated learning over multihop wireless networks with in-network aggregation, IEEE Trans. Wireless Commun. 21 (6) (2022) 4622–4634.
[13] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, et al., Federated learning for internet of things: A comprehensive survey, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1622–1658.
[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, et al., Towards federated learning at scale: System design, in: Proceedings of Machine Learning and Systems, Vol. 1, 2019, pp. 374–388.
[15] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-iid data, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3400–3413.
[16] L. Zhang, L. Shen, L. Ding, D. Tao, L.-Y. Duan, Fine-tuning global model via data-free knowledge distillation for non-iid federated learning, in: CVPR, 2022, pp. 10174–10183.
[17] J. Xu, B.S. Glicksberg, C. Su, P. Walker, et al., Federated learning for healthcare informatics, J. Healthc. Inform. Res. 5 (1) (2021) 1–19.
[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[19] C. Zhang, Y. Xie, H. Bai, B. Yu, et al., A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.
[20] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, et al., A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.
[21] X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions, ACM Comput. Surv. 54 (6) (2021) 1–36.
[22] V. Kulkarni, M. Kulkarni, A. Pant, Survey of personalization techniques for federated learning, in: 4th World Conference on Smart Trends in Systems, Security and Sustainability, IEEE, 2020, pp. 794–797.

[23] H. Zhu, J. Xu, S. Liu, Y. Jin, Federated learning on non-IID data: A survey, Neurocomputing 465 (2021) 371–390.
[24] M. Aledhari, R. Razzak, R.M. Parizi, F. Saeed, Federated learning: A survey on enabling technologies, protocols, and applications, IEEE Access 8 (2020) 140699–140725.
[25] D.C. Nguyen, Q.-V. Pham, P.N. Pathirana, M. Ding, et al., Federated learning for smart healthcare: A survey, ACM Comput. Surv. 55 (3) (2022) 1–37.
[26] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: Motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (6) (2020) 46–51.
[27] M.P. Sah, A. Singh, Aggregation techniques in federated learning: Comprehensive survey, challenges and opportunities, in: 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, IEEE, 2022, pp. 1962–1967.
[28] L.U. Khan, W. Saad, Z. Han, E. Hossain, et al., Federated learning for internet of things: Recent advances, taxonomy, and open challenges, IEEE Commun. Surv. Tutor. (2021).
[29] G. Antonella, F. Giancarlo, G. Gianluigi, M. Marcello, Data and model aggregation for radiomics applications: Emerging trend and open challenges, Inf. Fusion (2023) 101923.
[30] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University 33 (2004) (2004) 1–26.
[31] A.P. Siddaway, A.M. Wood, L.V. Hedges, How to do a systematic review: a best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses, Ann. Rev. Psychol. 70 (2019) 747–770.
[32] D. Moher, A. Liberati, J. Tetzlaff, D.G. Altman, PRISMA Group, Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement, Ann. Int. Med. 151 (4) (2009) 264–269.
[33] J. Wang, Q. Liu, H. Liang, G. Joshi, H.V. Poor, Tackling the objective inconsistency problem in heterogeneous federated optimization, NeurIPS 33 (2020) 7611–7623.
[34] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: ICML, 2020, pp. 5132–5143.
[35] B. Chai, K. Liu, R. Yang, Cross-domain federated data modeling on non-IID data, Comput. Intell. Neurosci. (2022).
[36] L.T. Nguyen, J. Kim, B. Shim, Gradual federated learning with simulated annealing, IEEE Trans. Signal Process. 69 (2021) 6299–6313.
[37] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, et al., Improving federated learning with quality-aware user incentive and auto-weighted model aggregation, IEEE Trans. Parallel Distrib. Syst. 33 (12) (2022) 4515–4529.
[38] Y. Mou, J. Geng, S. Welten, C. Rong, et al., Optimized federated learning on class-biased distributed data sources, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 146–158.
[39] L. Liu, J. Zhang, S. Song, K.B. Letaief, Communication-efficient federated distillation with active data sampling, in: ICC, IEEE, 2022, pp. 201–206.
[40] A. Kundu, P. Yu, L. Wynter, S.H. Lim, Robustness and personalization in federated learning: A unified approach via regularization, in: International Conference on Edge Computing and Communications, IEEE, 2022, pp. 1–11.
[41] S. Caldas, S.M.K. Duddu, P. Wu, T. Li, J. Konečnỳ, et al., Leaf: A benchmark for federated settings, 2018, arXiv preprint.
[42] H. Zhu, Y. Zhou, H. Qian, Y. Shi, et al., Online client selection for asynchronous federated learning with fairness consideration, IEEE Trans. Wireless Commun. (2022).
[43] X. Zhou, W. Liang, I. Kevin, K. Wang, et al., Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems, IEEE Wirel. Commun. 30 (2) (2023) 82–89.
[44] W. Zhang, T. Zhou, Q. Lu, X. Wang, et al., Dynamic-fusion-based federated learning for COVID-19 detection, IEEE Internet Things J. 8 (21) (2021) 15884–15891.
[45] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.
[46] Y. Chen, X. Qin, J. Wang, C. Yu, W. Gao, Fedhealth: A federated transfer learning framework for wearable healthcare, IEEE Intell. Syst. 35 (4) (2020) 83–93.
[47] W. Hou, J. Sun, G. Gui, T. Ohtsuki, et al., Federated learning for DL-CSI prediction in FDD massive MIMO systems, IEEE Wirel. Commun. Lett. 10 (8) (2021) 1810–1814.
[48] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, et al., Feddane: A federated newton-type method, in: 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2019, pp. 1227–1231.
[49] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, et al., Fetchsgd: Communication-efficient federated learning with sketching, in: ICML, 2020, pp. 8253–8265.
[50] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, Y. Chen, Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking, in: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, 2021, pp. 42–55.
[51] Q. Li, B. He, D. Song, Model-contrastive federated learning, in: CVPR, 2021, pp. 10713–10722.
[52] A. Fallah, A. Mokhtari, A. Ozdaglar, Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach, NeurIPS 33 (2020) 3557–3568.
[53] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: ICML, 2017, pp. 1126–1135.
[54] D.A.E. Acar, Y. Zhao, R. Matas, M. Mattina, et al., Federated learning based on dynamic regularization, in: ICLR, 2020, pp. 4615–4625.
[55] E. Sannara, F. Portet, P. Lalanda, V. German, A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison, in: International Conference on Pervasive Computing and Communications, IEEE, 2021, pp. 1–10.
[56] A. Mao, E. Huang, H. Gan, K. Liu, FedAAR: A novel federated learning framework for animal activity recognition with wearable sensors, Animals 12 (16) (2022) 2142.
[57] Y. Li, Y. Guo, M. Alazab, S. Chen, et al., Joint optimal quantization and aggregation of federated learning scheme in VANETs, IEEE Trans. Intell. Transp. Syst. (2022).
[58] J. Liu, J. Lou, L. Xiong, J. Liu, X. Meng, Projected federated averaging with heterogeneous differential privacy, Proc. VLDB Endow. 15 (4) (2021) 828–840.
[59] J. Bai, A. Sajjanhar, Y. Xiang, X. Tong, S. Zeng, Fedewa: Federated learning with elastic weighted averaging, in: IJCNN, IEEE, 2022, pp. 1–8.
[60] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, et al., Fair: Quality-aware federated learning with precise user incentive and model aggregation, in: Conference on Computer Communications, IEEE, 2021, pp. 1–10.
[61] M. Hong, S.-K. Kang, J.-H. Lee, Weighted averaging federated learning based on example forgetting events in label imbalanced non-IID, Appl. Sci. 12 (12) (2022) 5806.
[62] R. Ye, M. Xu, J. Wang, C. Xu, et al., FedDisco: Federated learning with discrepancy-aware collaboration, in: ICML, 2023.
[63] Z. Tang, F. Shao, L. Chen, Y. Ye, et al., Optimizing federated learning on non-IID data using local Shapley value, in: International Conference on Artificial Intelligence, Springer, 2021, pp. 164–175.
[64] H. Wu, P. Wang, Fast-convergent federated learning with adaptive weighting, IEEE Trans. Cogn. Commun. Netw. 7 (4) (2021) 1078–1088.
[65] Y. Wang, B. Kantarci, Reputation-enabled federated learning model aggregation in mobile platforms, in: ICC, IEEE, 2021, pp. 1–6.
[66] H.T. Nguyen, V. Sehwag, S. Hosseinalipour, C.G. Brinton, et al., Fast-convergent federated learning, IEEE J. Sel. Areas Commun. 39 (1) (2020) 201–218.
[67] D. Geng, H. He, X. Lan, C. Liu, Bearing fault diagnosis based on improved federated learning algorithm, Computing 104 (1) (2022) 1–19.
[68] S. Zhao, J. Liu, G. Ma, J. Yang, D. Liu, Z. Li, Two-phased federated learning with clustering and personalization for natural gas load forecasting, in: Trustworthy Federated Learning: 1st International Workshop, 13448, Springer Nature, 2023, p. 130.
[69] J.-h. Duan, W. Li, D. Zou, R. Li, S. Lu, Federated learning with data-agnostic distribution fusion, in: CVPR, 2023, pp. 8074–8083.
[70] S. Ji, S. Pan, G. Long, X. Li, et al., Learning private neural language modeling with attentive aggregation, in: IJCNN, IEEE, 2019, pp. 1–8.
[71] S. Li, T. Zhou, X. Tian, D. Tao, Learning to collaborate in decentralized learning of personalized models, in: CVPR, 2022, pp. 9766–9775.
[72] Z. Li, T. Lin, X. Shang, C. Wu, Revisiting weighted aggregation in federated learning with neural networks, in: ICML, 2023.
[73] Y. Mi, Y. Mu, S. Zhou, J. Guan, Fedmdr: Federated model distillation with robust aggregation, in: Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data, Springer, 2021, pp. 18–32.
[74] T. Lin, L. Kong, S.U. Stich, M. Jaggi, Ensemble distillation for robust model fusion in federated learning, NeurIPS 33 (2020) 2351–2363.
[75] C. Wu, F. Wu, L. Lyu, Y. Huang, X. Xie, Communication-efficient federated learning via knowledge distillation, Nat. Commun. 13 (1) (2022) 1–8.
[76] C. Wu, F. Zhang, F. Wu, Distributed modelling approaches for data privacy preserving, in: 5th International Conference on Multimedia Big Data, IEEE, 2019, pp. 357–365.
[77] C. He, M. Annavaram, S. Avestimehr, Group knowledge transfer: Federated learning of large cnns at the edge, NeurIPS 33 (2020) 14068–14080.
[78] W. Zhuang, Y. Wen, X. Zhang, X. Gan, et al., Performance optimization of federated person re-identification via benchmark analysis, in: 28th ACM International Conference on Multimedia, 2020, pp. 955–963.
[79] L. Hu, H. Yan, L. Li, Z. Pan, et al., MHAT: an efficient model-heterogenous aggregation training scheme for federated learning, Inform. Sci. 560 (2021) 493–503.
[80] X. Ni, X. Shen, H. Zhao, Federated optimization via knowledge codistillation, Expert Syst. Appl. 191 (2022) 116310.
[81] K. Ozkara, N. Singh, D. Data, S. Diggavi, QuPeD: Quantized personalization via distillation with applications to federated learning, NeurIPS 34 (2021) 3622–3634.
[82] X. Gong, A. Sharma, S. Karanam, Z. Wu, et al., Ensemble attention distillation for privacy-preserving federated learning, in: ICCV, 2021, pp. 15076–15086.

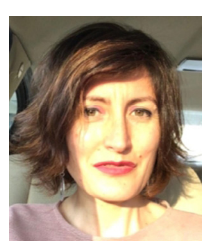
[83] C. Zhou, H. Tian, H. Zhang, J. Zhang, et al., TEA-fed: time-efficient asynchronous federated learning for edge computing, in: 18th ACM International Conference on Computing Frontiers, 2021, pp. 30–37.
[84] Y. Chen, X. Sun, Y. Jin, Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation, IEEE Trans. Neural Netw. Learn. Syst. 31 (10) (2019) 4229–4238.
[85] Y. Gao, L. Liu, X. Zheng, C. Zhang, H. Ma, Federated sensing: Edge-cloud elastic collaborative learning for intelligent sensing, IEEE Internet Things J. 8 (14) (2021) 11100–11111.
[86] Y. Jin, L. Jiao, M. Ji, Z. Qian, et al., Scheduling in-band network telemetry with convergence-preserving federated learning, IEEE/ACM Trans. Netw. (2023).
[87] Q. Wu, X. Chen, T. Ouyang, Z. Zhou, et al., Hiflash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association, IEEE Trans. Parallel Distrib. Syst. 34 (5) (2023) 1560–1579.
[88] Y. Gao, L. Liu, B. Hu, T. Lei, H. Ma, Federated region-learning for environment sensing in edge computing system, IEEE Trans. Netw. Sci. Eng. 7 (4) (2020) 2192–2204.
[89] Z. Chen, H. Cui, E. Wu, X. Yu, Dynamic asynchronous anti poisoning federated deep learning with blockchain-based reputation-aware solutions, Sensors 22 (2) (2022) 684.
[90] J. Lee, H. Ko, S. Pack, Adaptive deadline determination for mobile device selection in federated learning, IEEE Trans. Veh. Technol. 71 (3) (2021) 3367–3371.
[91] J. Liu, J.H. Wang, C. Rong, Y. Xu, et al., FedPA: An adaptively partial model aggregation strategy in federated learning, Comput. Netw. 199 (2021) 108468.
[92] Y. Sun, J. Shao, Y. Mao, J. Zhang, Asynchronous semi-decentralized federated edge learning for heterogeneous clients, in: ICC, IEEE, 2022, pp. 5196–5201.
[93] D. Huba, J. Nguyen, K. Malik, R. Zhu, et al., Papaya: Practical, private, and scalable federated learning, in: Proceedings of Machine Learning and Systems, Vol. 4, 2022, pp. 814–832.
[94] T. Huang, W. Lin, W. Wu, L. He, et al., An efficiency-boosting client selection scheme for federated learning with fairness guarantee, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2020) 1552–1564.
[95] J. Liu, H. Xu, L. Wang, Y. Xu, et al., Adaptive asynchronous federated learning in resource-constrained edge computing, IEEE Trans. Mob. Comput. (2021).
[96] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, et al., Federated learning with buffered asynchronous aggregation, in: International Conference on Artificial Intelligence and Statistics, 2022, pp. 3581–3607.
[97] B. Li, S. Chen, K. Yu, Model fusion from unauthorized clients in federated learning, Mathematics 10 (20) (2022) 3751.
[98] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, et al., Asynchronous federated learning over wireless communication networks, IEEE Trans. Wireless Commun. (2022).
[99] L. You, S. Liu, Y. Chang, C. Yuen, A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement, IEEE Internet Things J. 9 (23) (2022) 24199–24211.
[100] S. Liu, Q. Chen, L. You, Fed2A: Federated learning mechanism in asynchronous and adaptive modes, Electronics 11 (9) (2022) 1393.
[101] J. Du, N. Qin, D. Huang, Y. Zhang, X. Jia, An efficient federated learning framework for machinery fault diagnosis with improved model aggregation and local model training, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[102] R. Wang, F. Yan, L. Yu, C. Shen, et al., A federated transfer learning method with low-quality knowledge filtering and dynamic model aggregation for rolling bearing fault diagnosis, Mech. Syst. Signal Process. 198 (2023) 110413.
[103] E. Baccarelli, M. Scarpiniti, A. Momenzadeh, S.S. Ahrabi, AFAFed—Asynchronous fair adaptive federated learning for IoT stream applications, Comput. Commun. 195 (2022) 376–402.
[104] A. Sultana, M.M. Haque, L. Chen, F. Xu, X. Yuan, Eiffel: Efficient and fair scheduling in adaptive federated learning, IEEE Trans. Parallel Distrib. Syst. 33 (12) (2022) 4282–4294.
[105] C.-H. Hu, Z. Chen, E.G. Larsson, Device scheduling and update aggregation policies for asynchronous federated learning, in: 22nd International Workshop on Signal Processing Advances in Wireless Communications, IEEE, 2021, pp. 281–285.
[106] J. Xu, Y. Jin, W. Du, S. Gu, A federated data-driven evolutionary algorithm, Knowl.-Based Syst. 233 (2021) 107532.
[107] P. Li, Y. Zhao, L. Chen, K. Cheng, et al., Uncertainty measured active client selection for federated learning in smart grid, in: International Conference on Smart Internet of Things, IEEE, 2022, pp. 148–153.
[108] M.P. Uddin, Y. Xiang, X. Lu, J. Yearwood, L. Gao, Federated learning via disentangled information bottleneck, IEEE Trans. Serv. Comput. (2022).
[109] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, Budget-aware online control of edge federated learning on streaming data with stochastic inputs, IEEE J. Sel. Areas Commun. 39 (12) (2021) 3704–3722.
[110] V. Rey, P.M.S. Sánchez, A.H. Celdrán, G. Bovet, Federated learning for malware detection in iot devices, Comput. Netw. 204 (2022) 108693.
[111] M. Xue, W. Chenglin, An asynchronous quasi-cloud/edge/client collaborative federated learning mechanism for fault diagnosis, Chin. J. Electron. 30 (5) (2021) 969–977.
[112] S. Guo, B. Xiang, L. Chen, H. Yang, D. Yu, Multi-level federated learning mechanism with reinforcement learning optimizing in smart city, in: International Conference on Artificial Intelligence and Security, Springer, 2022, pp. 441–454.
[113] S. Ullah, D.-H. Kim, Federated learning using sparse-adaptive model selection for embedded edge computing, IEEE Access 9 (2021) 167868–167879.
[114] X. Wei, M. Hou, C. Ren, X. Li, H. Yue, MSSA-FL: High-performance multi-stage semi-asynchronous federated learning with non-IID data, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2022, pp. 172–187.
[115] Q. Ma, Y. Xu, H. Xu, Z. Jiang, et al., FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing, IEEE J. Sel. Areas Commun. 39 (12) (2021) 3654–3672.
[116] W. Wu, L. He, W. Lin, R. Mao, et al., SAFA: A semi-asynchronous protocol for fast federated learning with low overhead, IEEE Trans. Comput. 70 (5) (2020) 655–668.
[117] J. Hao, Y. Zhao, J. Zhang, Time efficient federated learning with semi-asynchronous communication, in: 26th International Conference on Parallel and Distributed Systems, IEEE, 2020, pp. 156–163.
[118] T.Q. Dinh, D.N. Nguyen, D.T. Hoang, P.T. Vu, E. Dutkiewicz, Enabling large-scale federated learning over wireless edge networks, in: Global Communications Conference, IEEE, 2021, pp. 01–06.
[119] S. Luo, X. Chen, Q. Wu, Z. Zhou, S. Yu, HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning, IEEE Trans. Wireless Commun. 19 (10) (2020) 6535–6548.
[120] L. Tang, Z. Wang, H. Pu, Z. Wu, Q. Chen, Research on efficient federated learning communication mechanism based on adaptive gradient compression, J. Electron. Inf. Technol. 45 (1) (2023) 227–234.
[121] Y. Deng, F. Lyu, J. Ren, Y. Zhang, et al., SHARE: Shaping data distribution at edge for communication-efficient hierarchical federated learning, in: 41st International Conference on Distributed Computing Systems, IEEE, 2021, pp. 24–34.
[122] T.Q. Dinh, D.N. Nguyen, D.T. Hoang, T.V. Pham, E. Dutkiewicz, In-network computation for large-scale federated learning over wireless edge networks, IEEE Trans. Mob. Comput. (2022).
[123] Y. Cai, W. Xi, Y. Shen, Y. Peng, et al., High-efficient hierarchical federated learning on non-IID data with progressive collaboration, Future Gener. Comput. Syst. 137 (2022) 111–128.
[124] M. Yemini, R. Saha, E. Ozfatura, D. Gündüz, A.J. Goldsmith, Semi-decentralized federated learning with collaborative relaying, in: International Symposium on Information Theory, IEEE, 2022, pp. 1471–1476.
[125] S. Hosseinalipour, S.S. Azam, C.G. Brinton, N. Michelusi, et al., Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks, IEEE/ACM Trans. Netw. 30 (4) (2022) 1569–1584.
[126] L. Yang, Y. Lu, J. Cao, J. Huang, M. Zhang, E-tree learning: A novel decentralized model learning framework for edge ai, IEEE Internet Things J. 8 (14) (2021) 11290–11304.
[127] Q. Cao, X. Zhang, Y. Zhang, Y. Zhu, Layered model aggregation based federated learning in mobile edge networks, in: ICC, IEEE, 2021, pp. 1–6.
[128] R. Song, L. Zhou, V. Lakshminarasimhan, A. Festag, et al., Federated learning framework coping with hierarchical heterogeneity in cooperative its, in: 25th International Conference on Intelligent Transportation Systems, IEEE, 2022, pp. 3502–3508.
[129] P.H. Mirzaee, M. Shojafar, H. Cruickshank, R. Tafazolli, CHFL: A collaborative hierarchical federated intrusion detection system for vehicular networks, in: 2022 IEEE Symposium on Computers and Communications, IEEE, 2022, pp. 1–7.
[130] S. Lonare, R. Bhramaramba, Model aggregation federated learning approach for vehicular traffic forecasting, J. Eng. Sci. Technol. Rev. 14 (3) (2021).
[131] C. Feng, H.H. Yang, D. Hu, Z. Zhao, et al., Mobility-aware cluster federated learning in hierarchical wireless networks, IEEE Trans. Wireless Commun. 21 (10) (2022) 8441–8458.
[132] W.Y.B. Lim, J.S. Ng, Z. Xiong, J. Jin, et al., Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning, IEEE Trans. Parallel Distrib. Syst. 33 (3) (2021) 536–550.
[133] D. Hui, L. Zhuo, C. Xin, Quality-aware incentive mechanism design based on matching game for hierarchical federated learning, in: Conference on Computer Communications Workshops, IEEE, 2022, pp. 1–6.
[134] A.A. Al-Saedi, E. Casalicchio, V. Boeva, An energy-aware multi-criteria federated learning model for edge computing, in: 8th International Conference on Future Internet of Things and Cloud, IEEE, 2021, pp. 134–143.
[135] Y. Shi, L. Shen, K. Wei, Y. Sun, et al., Improving the model consistency of decentralized federated learning, in: ICML, 2023.

[136] H. Cai, Y. Zhang, S. Wang, A. Zhao, M. Zhang, Trusted federated secure aggregation via similarity clustering, J. Electron. Inf. Technol. 45 (3) (2023) 1–11.
[137] D. Xiao, B. Cao, W. Wu, EFL-WP: Federated learning-based workload prediction in inter-cloud environments, in: IJCNN, IEEE, 2022, pp. 1–10.
[138] G. Long, M. Xie, T. Shen, T. Zhou, et al., Multi-center federated learning: clients clustering for better personalization, World Wide Web 26 (1) (2023) 481–500.
[139] F.P.-C. Lin, S. Hosseinalipour, S.S. Azam, C.G. Brinton, N. Michelusi, Semi-decentralized federated learning with cooperative D2D local model aggregations, IEEE J. Sel. Areas Commun. 39 (12) (2021) 3851–3869.
[140] C. Palihawadana, N. Wiratunga, A. Wijekoon, H. Kalutarage, FedSim: Similarity guided model aggregation for federated learning, Neurocomputing 483 (2022) 432–445.
[141] C. Briggs, Z. Fan, P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in: IJCNN, IEEE, 2020, pp. 1–9.
[142] D. Wang, N. Zhang, M. Tao, Adaptive clustering-based model aggregation for federated learning with imbalanced data, in: 22nd International Workshop on Signal Processing Advances in Wireless Communications, IEEE, 2021, pp. 591–595.
[143] D. Wang, N. Zhang, M. Tao, Clustered federated learning with weighted model aggregation for imbalanced data, China Commun. 19 (8) (2022) 41–56.
[144] Y. Sun, J. Shao, Y. Mao, J.H. Wang, J. Zhang, Semi-decentralized federated edge learning for fast convergence on non-iid data, in: Wireless Communications and Networking Conference, IEEE, 2022, pp. 1898–1903.
[145] Z. Yan, Y.Z. Yi, Z. JiLin, Z. NaiLiang, et al., Federated learning model training method based on data features perception aggregation, in: 94th Vehicular Technology Conference, IEEE, 2021, pp. 1–7.
[146] Z. Li, Z. Chen, X. Wei, S. Gao, et al., HPFL-CN: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering, in: 19th Annual IEEE International Conference on Sensing, Communication, and Networking, IEEE, 2022, pp. 325–333.
[147] W. Bao, H. Wang, J. Wu, J. He, Optimizing the collaboration structure in cross-silo federated learning, in: ICML, 2023.
[148] E. Isik-Polat, G. Polat, A. Kocyigit, ARFED: Attack-resistant federated averaging based on outlier elimination, Future Gener. Comput. Syst. 141 (2023) 626–650.
[149] W. Liu, X. Xu, D. Li, et al., Privacy preservation for federated learning with robust aggregation in edge computing, IEEE Internet Things J. 10 (8) (2023) 7343–7355.
[150] Y. Mao, X. Yuan, X. Zhao, S. Zhong, Romoa: Robust model aggregation for the resistance of federated learning to model poisoning attacks, in: European Symposium on Research in Computer Security, Springer, 2021, pp. 476–496.
[151] R. Xu, X. Feng, H. Zheng, Robust model aggregation for federated learning with heterogeneous clients, in: 7th International Conference on Computer and Communications, IEEE, 2021, pp. 1606–1610.
[152] W. Zhou, Y. Li, S. Chen, B. Ding, Real-time data processing architecture for multi-robots based on differential federated learning, in: SmartWorld, Ubiquitous Intelligence & Computing, IEEE, 2018, pp. 462–471.
[153] J. Zhang, B. Chen, S. Yu, H. Deng, PEFL: A privacy-enhanced federated learning scheme for big data analytics, in: Global Communications Conference, IEEE, 2019, pp. 1–6.
[154] P. Ramanan, K. Nakayama, Baffle: Blockchain based aggregator free federated learning, in: International Conference on Blockchain, IEEE, 2020, pp. 72–81.
[155] Y. Belal, A. Bellet, S.B. Mokhtar, V. Nitu, PEPPER: Empowering user-centric recommender systems over gossip learning, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6 (3) (2022) 1–27.
[156] Y. Wang, Z. Su, T.H. Luan, R. Li, K. Zhang, Federated learning with fair incentives and robust aggregation for UAV-aided crowdsensing, IEEE Trans. Netw. Sci. Eng. 9 (5) (2021) 3179–3196.
[157] X. He, H. Zhu, Q. Ling, Byzantine-robust and communication-efficient distributed non-convex learning over non-IID data, in: International Conference on Acoustics, Speech and Signal Processing, IEEE, 2022, pp. 5223–5227.
[158] Z. Liu, J. Guo, W. Yang, J. Fan, et al., Privacy-preserving aggregation in federated learning: A survey, IEEE Trans. Big Data (2022).
[159] W. Yang, B. Liu, C. Lu, N. Yu, Privacy preserving on updated parameters in federated learning, in: Proceedings of the ACM Turing Celebration Conference-China, 2020, pp. 27–31.
[160] F. Tang, W. Wu, J. Liu, H. Wang, M. Xian, Privacy-preserving distributed deep learning via homomorphic re-encryption, Electronics 8 (4) (2019) 411.
[161] D. Stripelis, H. Saleem, T. Ghai, N. Dhinagar, U. Gupta, C. Anastasiou, et al., Secure neuroimaging analysis using federated learning with homomorphic encryption, in: 17th International Symposium on Medical Information Processing and Analysis, Vol. 12088, SPIE, 2021, pp. 351–359.
[162] D. Xu, S. Yuan, X. Wu, Achieving differential privacy in vertically partitioned multiparty learning, in: International Conference on Big Data, IEEE, 2021, pp. 5474–5483.
[163] C. Fang, Y. Guo, Y. Hu, B. Ma, L. Feng, A. Yin, Privacy-preserving and communication-efficient federated learning in Internet of Things, Comput. Secur. 103 (2021) 102199.
[164] Y. Cheng, Y. Liu, T. Chen, Q. Yang, Federated learning for privacy-preserving AI, Commun. ACM 63 (12) (2020) 33–36.
[165] K. Mandal, G. Gong, PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks, in: Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop, 2019, pp. 57–68.
[166] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, Q. Yang, Privacy-preserving heterogeneous federated transfer learning, in: International Conference on Big Data, IEEE, 2019, pp. 2552–2559.
[167] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intell. Syst. 35 (4) (2020) 70–82.
[168] Y. Li, H. Li, G. Xu, X. Huang, R. Lu, Efficient privacy-preserving federated learning with unreliable users, IEEE Internet Things J. 9 (13) (2021) 11590–11603.
[169] J. Ma, S.-A. Naas, S. Sigg, X. Lyu, Privacy-preserving federated learning based on multi-key homomorphic encryption, Int. J. Intell. Syst. 37 (9) (2022) 5880–5901.
[170] H. Zhu, R. Wang, Y. Jin, K. Liang, J. Ning, Distributed additive encryption and quantization for privacy preserving federated deep learning, Neurocomputing 463 (2021) 309–327.
[171] D. Boer, S. Kramer, Secure sum outperforms homomorphic encryption in (current) collaborative deep learning, 2020, arXiv preprint.
[172] E. Sotthiwat, L. Zhen, Z. Li, C. Zhang, Partially encrypted multi-party computation for federated learning, in: 21st International Symposium on Cluster, Cloud and Internet Computing, IEEE, 2021, pp. 828–835.
[173] S. Kadhe, N. Rajaraman, O.O. Koyluoglu, K. Ramchandran, Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning, 2020, arXiv preprint.
[174] C. Brunetta, G. Tsaloli, B. Liang, G. Banegas, A. Mitrokotsa, Non-interactive, secure verifiable aggregation for decentralized, privacy-preserving learning, in: Information Security and Privacy: 26th Australasian Conference, Springer, 2021, pp. 510–528.
[175] A. Fu, X. Zhang, N. Xiong, Y. Gao, et al., VFL: A verifiable federated learning with privacy-preserving for big data in industrial IoT, IEEE Trans. Ind. Inform. 18 (5) (2020) 3316–3326.
[176] R. Kanagavelu, Z. Li, J. Samsudin, Y. Yang, et al., Two-phase multi-party computation enabled privacy-preserving federated learning, in: International Symposium on Cluster, Cloud and Internet Computing, IEEE, 2020, pp. 410–419.
[177] H. Zhu, R.S.M. Goh, W.-K. Ng, Privacy-preserving weighted federated learning within the secret sharing framework, IEEE Access 8 (2020) 198275–198284.
[178] Y. Xu, C. Peng, W. Tan, Y. Tian, et al., Non-interactive verifiable privacy-preserving federated learning, Future Gener. Comput. Syst. 128 (2022) 365–380.
[179] G. Xu, H. Li, Y. Zhang, S. Xu, et al., Privacy-preserving federated deep learning with irregular users, IEEE Trans. Dependable Secure Comput. 19 (2) (2020) 1364–1381.
[180] H. Chen, H. Li, G. Xu, Y. Zhang, X. Luo, Achieving privacy-preserving federated learning with irrelevant updates over e-health applications, in: ICC, IEEE, 2020, pp. 1–6.
[181] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, et al., Blockchain challenges and opportunities: A survey, 14, (4) 2018, pp. 352–375.
[182] F. Casino, T.K. Dasaklis, C. Patsakis, A systematic literature review of blockchain-based applications: Current status, classification and open issues, Telemat. Inform. 36 (2019) 55–81.
[183] D. Hamouda, M.A. Ferrag, N. Benhamida, H. Seridi, PPSS: A privacy-preserving secure framework using blockchain-enabled federated deep learning for industrial IoTs, Pervasive Mob. Comput. 88 (2023) 101738.
[184] A.P. Kalapaaking, I. Khalil, M.S. Rahman, M. Atiquzzaman, et al., Blockchain-based federated learning with secure aggregation in trusted execution environment for Internet-of-Things, IEEE Trans. Ind. Inform. 19 (2) (2022) 1703–1714.
[185] J. Qi, F. Lin, Z. Chen, C. Tang, et al., High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation, IEEE Internet Things J. 9 (19) (2022) 18378–18391.
[186] X. Chen, T. Wang, S. Zhang, The design of reputation system for blockchain-based federated learning, in: International Conference on Artificial Intelligence and Blockchain Technology, IEEE, 2021, pp. 114–120.
[187] T. Ranathunga, A. McGibney, S. Rea, S. Bharti, Blockchain based decentralised model aggregation for cross-silo federated learning in industry 4.0, IEEE Internet Things J. (2022).
[188] Global Platform, Introduction to trusted execution environments, Global Platform (2018).
[189] L. Zhao, J. Jiang, B. Feng, Q. Wang, et al., Sear: Secure and efficient aggregation for byzantine-robust federated learning, IEEE Trans. Dependable Secure Comput. 19 (5) (2021) 3329–3342.
[190] F. Mo, H. Haddadi, Efficient and private federated learning using tee, in: Proc. EuroSys Conf., Dresden, Germany, 2019.

[191] Y. Zhang, Z. Wang, J. Cao, R. Hou, D. Meng, ShuffleFL: gradient-preserving federated learning using trusted execution environment, in: Proceedings of the 18th ACM International Conference on Computing Frontiers, 2021, pp. 161–168.
[192] E. Kuznetsov, Y. Chen, M. Zhao, Securefl: Privacy preserving federated learning with sgx and trustzone, in: 2021 IEEE/ACM Symposium on Edge Computing, IEEE, 2021, pp. 55–67.
[193] F. Mo, H. Haddadi, K. Katevas, et al., PPFL: privacy-preserving federated learning with trusted execution environments, in: 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 94–108.
[194] M. Mohri, G. Sivek, A.T. Suresh, Agnostic federated learning, in: ICML, 2019, pp. 4615–4625.
[195] S.-M. Lee, J.-L. Wu, FedUA: An uncertainty-aware distillation-based federated learning scheme for image classification, Information 14 (4) (2023) 234.
[196] Y. Li, H. Yu, Y. Zeng, Q. Pan, HFSA: A semi-asynchronous hierarchical federated recommendation system in smart city, IEEE Internet Things J. (2023).
[197] Y. Zhang, D. Liu, M. Duan, L. Li, et al., FedMDS: An efficient model discrepancy-aware semi-asynchronous clustered federated learning framework, IEEE Trans. Parallel Distrib. Syst. 34 (3) (2023) 1007–1019.
[198] P. Qi, D. Chiaro, F. Piccialli, FL-FD: Federated learning-based fall detection with multimodal data fusion, Inf. Fusion (2023) 101890.
[199] M. Chetoui, M.A. Akhloufi, Peer-to-peer federated learning for COVID-19 detection using transformers, Computers 12 (5) (2023) 106.
[200] W. Zhang, Z. Wang, X. Li, Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis, Reliab. Eng. Syst. Saf. 229 (2023) 108885.
[201] H. Deng, J. Hu, R. Sharma, M. Mo, et al., NVAS: A non-interactive verifiable federated learning aggregation scheme for COVID-19 based on game theory, Comput. Commun. 206 (2023) 1–9.
[202] J. Li, Y. Meng, L. Ma, S. Du, et al., A federated learning based privacy-preserving smart healthcare system, IEEE Trans. Ind. Inform. 18 (3) (2021).
[203] A. Kumar, V. Purohit, V. Bharti, R. Singh, S.K. Singh, MediSecFed: Private and secure medical image classification in the presence of malicious clients, IEEE Trans. Ind. Inform. 18 (8) (2021) 5648–5657.
[204] O. Choudhury, Y. Park, T. Salonidis, A. Gkoulalas-Divanis, et al., Predicting adverse drug reactions on distributed health data using federated learning, in: AMIA Annual Symposium Proceedings, Vol. 2019, American Medical Informatics Association, 2019, p. 313.
[205] Z. Guo, L. You, S. Liu, J. He, B. Zuo, ICMFed: An incremental and cost-efficient mechanism of federated meta-learning for driver distraction detection, Mathematics 11 (8) (2023) 1867.
[206] H. Zhang, J. Bosch, H.H. Olsson, Real-time end-to-end federated learning: An automotive case study, in: 45th Annual Computers, Software, and Applications Conference, IEEE, 2021, pp. 459–468.
[207] X. Zhou, W. Liang, J. She, Z. Yan, et al., Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles, IEEE Trans. Veh. Technol. 70 (6) (2021) 5308–5317.
[208] Z. Zhang, S. Wang, Y. Hong, L. Zhou, Q. Hao, Distributed dynamic map fusion via federated learning for intelligent networked vehicles, in: International Conference on Robotics and Automation, IEEE, 2021, pp. 953–959.
[209] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: A selective model aggregation approach, IEEE Access 8 (2020) 23920–23935.
[210] S. Liu, J. Yu, X. Deng, S. Wan, FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks, IEEE Trans. Intell. Transp. Syst. 23 (2) (2021) 1616–1629.
[211] P. Wu, T. Imbiriba, J. Park, S. Kim, P. Closas, Personalized federated learning over non-IID data for indoor localization, in: 22nd International Workshop on Signal Processing Advances in Wireless Communications, IEEE, 2021, pp. 421–425.
[212] G. Huang, C. Wu, Y. Hu, C. Guo, Serverless distributed learning for smart grid analytics, Chin. Phys. B 30 (8) (2021) 088802.
[213] M. Imran, H. Yin, T. Chen, Q.V.H. Nguyen, et al., ReFRS: Resource-efficient federated recommender system for dynamic and diversified user preferences, ACM Trans. Inf. Syst. 41 (3) (2023) 1–30.
[214] M.S. Farooq, R. Tehseen, J.N. Qureshi, U. Omer, et al., FFM: Flood forecasting model using federated learning, IEEE Access 11 (2023) 24472–24483.
[215] A. Imteaj, M.H. Amini, Leveraging asynchronous federated learning to predict customers financial distress, Intell. Syst. Appl. 14 (2022) 200064.
[216] J. Li, X. Tong, J. Liu, L. Cheng, An efficient federated learning system for network intrusion detection, IEEE Syst. J. (2023).
[217] Y. Qin, M. Li, J. Zhu, Privacy-preserving federated learning framework in multimedia courses recommendation, Wirel. Netw. 29 (4) (2023) 1535–1544.
[218] S. Su, B. Li, X. Xue, One-shot federated learning without server-side training, Neural Netw. 164 (2023) 203–215.
[219] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, et al., Bayesian nonparametric federated learning of neural networks, in: ICML, 2019, pp. 7252–7261.
[220] S. Claici, M. Yurochkin, S. Ghosh, J. Solomon, Model fusion with Kullback-Leibler divergence, in: ICML, 2020, pp. 2038–2047.
[221] S. Shukla, N. Srivastava, Federated matched averaging with information-gain based parameter sampling, in: 1st International Conference on AI-ML-Systems, 2021, pp. 1–7.
[222] W.J. Yun, Y. Kwak, H. Baek, S. Jung, et al., SlimFL: Federated learning with superposition coding over slimmable neural networks, IEEE/ACM Trans. Netw. (2023).
[223] P. Kairouz, H.B. McMahan, B. Avent, et al., Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (1–2) (2021) 1–210.
[224] W. Ni, Y. Liu, Z. Yang, H. Tian, X. Shen, Federated learning in multi-RIS-aided systems, IEEE Internet Things J. 9 (12) (2021) 9608–9624.
[225] Z. Lin, H. Liu, Y.-J.A. Zhang, Relay-assisted cooperative federated learning, IEEE Trans. Wireless Commun. 21 (9) (2022) 7148–7164.
[226] K.B. Letaief, W. Chen, Y. Shi, J. Zhang, et al., The roadmap to 6G: AI empowered wireless networks, IEEE Commun. Mag. 57 (8) (2019) 84–90.
[227] X. Xiao, Z. Tang, C. Li, B. Xiao, K. Li, SCA: sybil-based collusion attacks of iIoT data poisoning in federated learning, IEEE Trans. Ind. Inform. 19 (3) (2023) 2608–2618.
[228] S. Wang, BlockFedML: Blockchained federated machine learning systems, in: International Conference on Intelligent Computing, Automation and Systems, IEEE, 2019, pp. 751–756.
[229] Z. Chen, P. Tian, W. Liao, W. Yu, Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning, IEEE Trans. Netw. Sci. Eng. 8 (2) (2020) 1070–1083.

Pian Qi is a Ph.D. student in Mathematics and Computer Science at the University of Naples Federico II. She received the Master Degree in Civil Engineering at the University of Geosciences, Beijing, China. Her research interests are focused on Deep Learning methodologies, Graph Mining and Data Mining techniques.

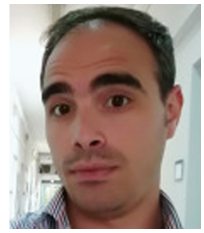
Diletta Chiaro is a Ph.D. student in Mathematics and Computer Science at the University of Naples Federico II. She received the Master Degree in Statistical Sciences at the University of Naples Federico II in 2021. Her research interests are focused on Machine Learning methodologies, Graph Mining and Data Visualization techniques.



Antonella Guzzo received the Ph.D. degree in computer and systems engineering from the University of Calabria, Italy, in 2005. Currently, she is an associate professor of computer science in the DIMES Department, University of Calabria. Previously, she was a research fellow in the High Performance Computing and Networks Institute (ICAR-CNR), National Research Council, Italy. Her research interests include process mining, data mining, and knowledge representation. She is a member of the IEEE.

Michele Ianni is an Assistant Professor at the Department of Computer Science, Modeling, Electronic and System Engineering (DIMES) of the University of Calabria, Italy. He received the Ph.D. degree in Information and Communication Technologies from the University of Calabria, Italy, in 2018. During his Ph.D. he was a visiting researcher in SecLab, University of California, Santa Barbara. Previously, He was a Postdoctoral Researcher at the University of Calabria, Italy and at the University of Verona, Italy. His main research interests include Binary Analysis and Exploitation, Obfuscation, Watermarking, Malware, Trusted Execution Environments and IoT Security.

Giancarlo Fortino (IEEE Fellow 2022) is Full Professor of Computer Engineering at the Dept of Informatics, Modeling, Electronics, and Systems of the University of Calabria (Unical), Italy. He received a Ph.D. in Computer Engineering from Unical in 2000. He is also distinguished professor at Wuhan University of Technology and Huazhong Agricultural University (China), high-end expert at HUST (China), senior research fellow at the Italian ICAR-CNR Institute, CAS PIFI visiting scientist at SIAT – Shenzhen, high-end expert of the Henan province, and Distinguished Lecturer for IEEE Sensors Council (2021–23). At Unical, he is the Rector's delegate to Int'l relations, the chair of the Ph.D. School in ICT , the director of the Postgraduate Master course in INTER-IoT, and the director of the SPEME lab as well as co-chair of Joint labs on IoT established between Unical and WUT, SMU and HZAU Chinese universities, respectively. Fortino is currently the scientific responsible of the Digital Health group of the Italian CINI National Laboratory at Unical. He is Highly Cited Researcher 2020–2022 in Computer Science by Clarivate. Currently he has 20 highly cited papers in WoS, and h-index=73 with 19000+ citations in Google Scholar. His research interests include wearable computing systems, e-Health, Internet of Things, and agent-based computing. He is author of 600+ papers in int'l journals, conferences and books. He is (founding) series editor of IEEE Press Book Series on Human–Machine Systems and EiC of Springer Internet of Things series and AE of premier int'l journals such as IEEE TASE (senior editor), IEEE TAFFC-CS, IEEE THMS, IEEE T-AI, IEEE IoTJ, IEEE SJ, IEEE JBHI, IEEE SMCM, IEEE OJEMB, IEEE OJCS, Information Fusion, EAAI, etc. He chaired many int'l workshops and conferences (120+), was involved in a huge number of int'l conferences/workshops (500+) as IPC member, is/was guest-editor of many special issues (75+). He is cofounder and CEO of SenSysCal S.r.l., a Unical spinoff focused on innovative IoT systems, and recently cofounder and vice-CEO of the spin-off Bigtech S.r.l, focused on big data, AI and IoT technologies. Fortino is currently member of the IEEE SMCS BoG and of the IEEE Press BoG, and former chair of the IEEE SMCS Italian Chapter.



Francesco Piccialli is currently Associate Professor of Computer Science and Artificial Intelligence at the Department of Mathematics and Applications "R. Caccioppoli" (DMA), University of Naples Federico II (UNINA), Italy. He received a Laurea Degree (BSc+MSc) in Computer Science and the Ph.D. in Computational and Computer sciences at the University of Naples Federico II. He is the head and co-founder of the M.O.D.A.L. research group that is engaged in cutting-edge on novel methodologies, architectures and applications in the Artificial Intelligence, Machine and Deep Learning fields also focusing on their emerging application domains. He is also research fellow at C.I.N.I. (National Interuniversity Consortium for Informatics) from 2013. He has been involved in research and development projects in the research areas of Artificial Intelligence, Data Science and Internet of Things. He is author of many scientific papers (130+) in international conferences and top-ranking journals (IEEE, ACM, Springer, Elsevier, Wiley, Nature). Currently he serves as Associate Editor of many prestigious, international and well-recognized journals.