

Review

Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey

Muhammad Asad ^{*}, Saima Shaukat, Dou Hu, Zekun Wang, Ehsan Javanmardi , Jin Nakazato 
and Manabu Tsukada 

Graduate School of Information Science and Technology, Department of Creative Informatics,
The University of Tokyo, Tokyo 113-8654, Japan; saima@g.ecc.u-tokyo.ac.jp (S.S.);
douhu@g.ecc.u-tokyo.ac.jp (D.H.); wangzekun.felix@gmail.com (Z.W.); ejavanmardi@g.ecc.u-tokyo.ac.jp (E.J.);
jin-nakazato@g.ecc.u-tokyo.ac.jp (J.N.); mtsukada@g.ecc.u-tokyo.ac.jp (M.T.)

* Correspondence: asad@g.ecc.u-tokyo.ac.jp

Abstract: This paper explores the potential for communication-efficient federated learning (FL) in modern distributed systems. FL is an emerging distributed machine learning technique that allows for the distributed training of a single machine learning model across multiple geographically distributed clients. This paper surveys the various approaches to communication-efficient FL, including model updates, compression techniques, resource management for the edge and cloud, and client selection. We also review the various optimization techniques associated with communication-efficient FL, such as compression schemes and structured updates. Finally, we highlight the current research challenges and discuss the potential future directions for communication-efficient FL.

Keywords: federated learning; communication efficient; model compression; resource management; client selection; structured updates



Citation: Asad, M.; Shaukat, S.; Hu, D.; Wang, Z.; Javanmardi, E.; Nakazato, J.; Tsukada, M. Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey. *Sensors* **2023**, *23*, 7358. <https://doi.org/10.3390/s23177358>

Academic Editor: Josip Lorincz

Received: 31 July 2023

Revised: 19 August 2023

Accepted: 20 August 2023

Published: 23 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Federated learning (FL) is a rapidly growing field that enables multiple clients to train a machine learning model while preserving their data privacy [1]. It has been extensively used in various fields, including healthcare, finance, and social media, where privacy is critical [2,3]. In FL, the clients (devices) perform local training on their respective datasets and then share only the model updates with the server, aggregating the updates to generate a global model [4]. In contrast to conventional centralized machine learning, federated learning (FL) and distributed machine learning offer unique mechanisms for training models across decentralized devices or data sources. However, it is FL that emphasizes collaborative model training with a keen focus on preserving data privacy, minimizing communication overheads, and catering to dynamic and potentially heterogeneous data environments. Distributed machine learning, while reducing computational constraints through parallel processing, often involves more frequent data exchanges without the inherent privacy-preserving design of FL [5]. Figure 1 provides a comprehensive architectural comparison of FL against the traditional centralized and distributed machine learning frameworks. It showcases the nuances in data distribution, model updates, and communication patterns among these paradigms, thus emphasizing the distinct attributes and advantages of FL.

FL is a revolutionary approach in distributed machine learning, promising enhanced data privacy and decentralized model training [6]. However, the communication overheads associated with FL have emerged as a considerable challenge, especially when they pertain to scalability and efficiency. In the FL paradigm, communication costs are bifurcated into two primary categories: upload and download costs. The former encapsulates the data transmitted by clients to the server during the training phase, while the latter accounts for data fetched from the server by the clients [7]. Notably, these costs are influenced by

various parameters, such as the dataset size, model intricacy, client count, and network bandwidth [8,9].

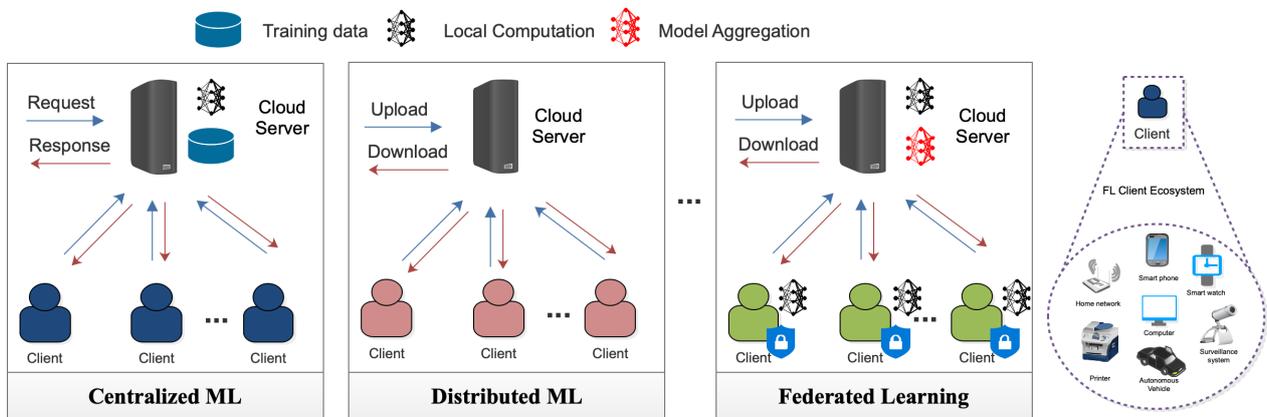


Figure 1. Comparison of FL with conventional centralized machine learning and distributed learning.

The ramifications of these communication costs on FL's efficiency have been the subject of intensive research. Empirical analyses have ascertained that communication costs can act as substantial constraints, inhibiting the scalability and effectiveness of FL systems [10,11]. To combat these challenges, innovative solutions, such as compression methodologies and model quantization, have been postulated [12,13]. Compression solutions primarily focus on diminishing the magnitude of model updates, thus curtailing upload expenses, whereas model quantization optimizes model parameters' precision, facilitating further reductions in upload costs.

Moreover, the aspiration for energy-efficient communication systems is a pressing concern that complements the drive for communication efficiency [14]. Reducing communication overheads through FL directly ties into energy savings. Given that every data exchange involves energy consumption, optimizing the FL process impacts the bandwidth and potentially contributes to reduced energy expenditures, a vital consideration for modern communication networks [15,16]. FL's capacity to minimize data transmission inherently reduces energy consumption, placing it at the forefront of strategies to develop energy-efficient communication networks.

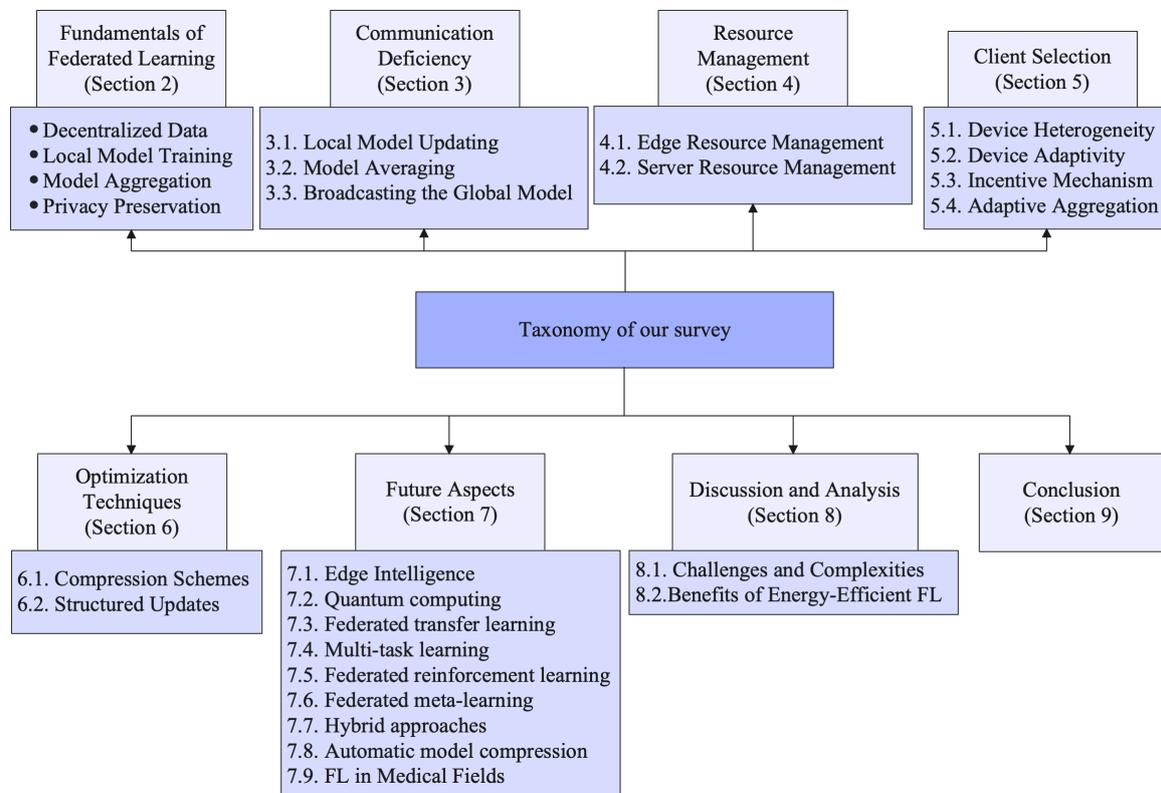
This paper provides an overview of the various communication-efficient FL strategies, including model updating, compression techniques, resource management for the edge and cloud, and client selection. An in-depth look at the different optimization techniques related to communication-efficient FL, such as compression schemes and structured updates, is also included. The potential of communication-efficient FL for emerging distributed applications is discussed, including the benefits and challenges that could arise from its integration into existing distributed systems.

The primary objective of this paper is to analyze the communication efficiency of FL and its impact on system performance. Thus, our following discussion will solely revolve around communication cost, disregarding other aspects of FL. Moreover, to the best of our knowledge, no previous survey has specifically focused on examining the communication cost of FL. Table 1 presents the existing work on FL relevant to our survey.

The rest of this paper is organized as follows: Section 2 presents the fundamentals of FL. Section 3 explains the communication deficiency in detail. Section 4 provides the details of resource management strategies in FL. Section 5 presents the importance of client selection in FL. Section 6 presents the optimization techniques of FL. Section 7 presents the potential future directions of FL regarding communication costs. Section 8 provides the discussion and analysis of this survey. Finally, Section 9 concludes this survey paper. A complete overview of this survey is presented in Figure 2.

Table 1. Existing surveys and their primary focus.

Reference	Year	Focus	Communication Constraints	Challenges
[1]	2021	Characteristics and the current practical application of FL	Yes	Network heterogeneity
[17]	2023	Threats and vulnerabilities of FL	No	Adversarial attacks
[18]	2021	Categorization of FL	Partially discussed	Design factors
[3]	2020	Comparison of different ML deployment architectures and in-depth investigation on FL	Partially discussed	Architectural robustness
[19]	2021	Advances and open challenges of FL	No	Privacy and communication
[20]	2021	Characteristics of edge FL	Yes	Security and privacy
[21]	2021	Non-identical and non-independent data distribution in FL	Partially	Communication efficiency
[22]	2022	FL in smart healthcare	No	Design factors
[23]	2023	Blockchain empowered FL	No	Privacy and security
[24]	2022	Security aspects of FL	No	Privacy and security
[25]	2022	Implementation of FL in centralized, decentralized, and heterogeneous approach	Partially discussed	Network heterogeneity
[26]	2022	Integration of FL with industrial IoT	No	Privacy preservation
[27]	2023	FL in wireless networks	Yes	High communication costs
[28]	2023	Review of existing studies on communication constraints in FL	Yes	Communication costs
[29]	2023	Threats to and flaws in the FL strategy	No	Privacy and Security
[30,31]	2020	FL in mobile edge computing	Partially discussed	Design factors
[32]	2020	Personalization of FL	No	Client selection

**Figure 2.** An overview of our survey.

2. Fundamentals of Federated Learning

In this section, we will explore the fundamentals of FL, including the decentralized nature of the data, local model training, model aggregation, and privacy preservation. Understanding these fundamentals is essential for understanding the challenges and opportunities associated with FL and developing approaches to reducing communication costs and improving performance. Table 2 provides a description of the fundamentals of FL, including the advantages and challenges associated with FL. The fundamentals are the following:

- **Decentralized data:** FL involves multiple clients or devices that hold their respective data. As a result, the data are decentralized and not stored in a central location [32,33]. This decentralized nature of data in FL helps preserve the local data's privacy, but it can also lead to increased communication costs [34]. The decentralized data distribution means more data must be transferred between the clients and the central server during the training process, leading to higher communication costs [35].
- **Local model training:** FL allows each client to perform local model training on its respective data. This local training ensures that the privacy of the local data is preserved, but it can also lead to increased communication costs [36]. The local model updates need to be sent to the central server, which aggregates them to generate a global model. The communication costs of sending these updates to the central server can be significant, particularly when the number of clients or data size is large [37,38].
- **Model aggregation:** After the local model training is completed, the clients send their model updates to the central server for aggregation [39,40]. The server aggregates the model updates to generate a global model, which reflects the characteristics of the data from all the clients [41]. The model aggregation process can lead to significant communication costs, particularly when the size of the model updates is large or the number of clients is high [22,42,43].
- **Privacy preservation:** FL is designed to preserve the privacy of the local data, but it can also lead to increased communication costs [44,45]. The privacy-preserving nature of FL means that the local data remain on the clients, and only the model updates are shared with the central server [46]. However, this also means more data must be transferred between the clients and the server during the training process, leading to higher communication costs.

Table 2. Fundamentals of FL.

Category	Description
Definition	FL is a machine learning setting where the goal is to train a model across multiple decentralized edge devices or servers holding local data samples, without explicitly exchanging data samples.
Key Components	The main elements of FL include the client devices holding local data, the central server that coordinates the learning process, and the machine learning models being trained.
Workflow	The typical FL cycle is as follows: (1) The server initializes the model and sends it to the clients; (2) Each client trains the model locally using its data; (3) The clients send their locally updated models or gradients to the server; (4) The server aggregates the received models (typically by averaging); (5) Steps 2–4 are repeated until convergence.
Advantages	The benefits of FL include (1) privacy preservation, as raw data remain on the client; (2) a reduction in bandwidth usage, as only model updates are transferred, not the data; (3) the potential for personalized models, as models can learn from local data patterns.
Challenges	FL faces several challenges, including (1) communication efficiency; (2) heterogeneity in terms of computation and data distribution across clients; (3) statistical challenges due to non-iid data; (4) privacy and security concerns.

Table 2. *Cont.*

Category	Description
Communication Efficiency Techniques	Communication efficiency can be improved using techniques, such as (1) federated averaging, which reduces the number of communication rounds; (2) model compression techniques, which reduce the size of model updates; (3) the use of parameter quantization or sparsification.
Data Distribution	In FL, data are typically distributed in a non-iid manner across clients due to the nature of edge devices. This unique distribution can lead to statistical challenges and influence the final model's performance.
Evaluation Metrics	Evaluation of FL models considers several metrics: (1) global accuracy, measuring how well the model performs on the entire data distribution; (2) local accuracy, measuring performance on individual client's data; (3) communication rounds, indicating the number of training iterations; (4) data efficiency, which considers the amount of data needed to reach a certain level of accuracy.

3. Communication Deficiency

The communication deficiency of FL is an important issue that needs to be addressed for this type of distributed machine learning to be successful. In FL, each client, typically a mobile device, must communicate with a centralized server to send and receive updates to the model [47]. As the number of clients increases, the amount of communication between the server and clients increases exponentially. This can become a major bottleneck, causing the training process to be slow and inefficient. Additionally, communication can be expensive, especially for mobile devices, so minimizing the amount of communication required for FL is important [48]. Figure 3 delves into the intricacies of the FL communication protocol. Beyond merely illustrating the flow, it captures the iterative nature of client–server interactions, highlighting the stages where communication overheads might arise and emphasizing the importance of efficient data exchanges in the FL process. The following section will examine the communication deficiency concerning local model updating and decentralized data training. In Table 3, we highlight the overview of some existing studies on the communication deficiency of FL.

Table 3. Existing research focusing on communication deficiency in FL.

Reference	Focus	Overview
[49]	Client selection	The algorithm recognizes the non-IID degrees of clients and chooses those with lower degrees of non-IID data to train the models with higher frequency.
[50]	Client selection	Optimizes the trade-off between maximizing the number of selected clients and minimizes the energy drawn from batteries for the selected clients in FL.
[51]	Resource management	The study uses cluster heads to communicate with the cloud server through edge aggregation, where clients upload their local models to their respective cluster heads. A joint communication and computation resource management scheme is also formulated through efficient client selection to achieve global cost minimization.
[52]	Client selection	The study divides clients into tiers based on their training performance. It selects clients from the same tier in each training round to mitigate the straggler problem. It employs an adaptive tier selection approach to update the tiering on the fly based on the observed training performance and accuracy.
[53]	Communication efficiency	The paper proposes the "In-Edge AI" framework that integrates deep reinforcement learning and FL with mobile edge systems in order to optimize mobile edge computing, caching, and communication.
[54]	Edge resource management	The study proposes a DTWN model and designs an edge association problem armed with FL. A multi-agent deep reinforcement learning-based algorithm is proposed to solve the problem. In addition, the study considers an edge association and communication resource allocation problem to minimize communication costs.

Table 3. Cont.

Reference	Focus	Overview
[55]	Edge resource management	The paper proposes a framework called concurrent federated reinforcement learning. Specifically, it protects the privacy of both the server and the edge node with the assistance of blockchain.
[56]	Edge resource management	The paper proposed an FL framework, which can securely update the data with the help of parallel blockchains. It considers a two-phase commit protocol and defines an auction scheme based on ML for price optimization.
[57]	Incentive mechanism	The paper considers a framework of a privacy-preserving incentive mechanism for encouraging the users to join the network. Specifically, the paper makes an extremely rigorous convergence analysis and derives a set of optimal contracts under the constraints of security demands and budget costs for each worker in the scenario.
[58]	Structured updates	The study shows an FL framework for autonomous driving. With the help of MEC nodes and blockchain, the system can achieve a lower latency and more accurate results between the vehicles, even if there are malicious vehicles and MEC nodes.
[59]	Incentive mechanism	The paper proposes an FL-based autonomous vehicle controller. To explain it deeper, the study uses a contract-theoretic incentive mechanism to speed up the process. It considers optimization methods to decrease the communication and computation cost for the system.
[60]	Incentive mechanism	The paper proposes a coded FL method that is based on an evolutionary game and a deep learning method to allocate the resource intelligently. The results show that the study mitigates the overall system computation and communication latency.
[61]	Optimization technique	The paper designs a client–edge–cloud hierarchical FL architecture. It develops an HierFAVG algorithm to allow edge servers to aggregate models partially to gain a higher efficiency.
[62]	Client selection	The study proposes a two-level hierarchical FL framework and designs two incentive mechanisms for resource allocation. The cluster selection mechanism of workers is based on an evolutionary game, and one deep-learning-based auction mechanism is designed for the model owner’s selection of cluster heads.
[63]	Resource management	The paper considers a maximum model accuracy problem of the wireless FL under the limited training time and latency constraint. It proposed a joint device scheduling and resource allocation policy.
[64]	Client selection	The study presents a Clients’ Eligibility Protocol (CEP) to work with heterogeneous clients in practical industrial scenarios efficiently. The CEP uses a trusted authority to calculate the client’s eligibility score based on local computing resources, such as the bandwidth, memory, and battery life, and selects the resourceful clients for training.

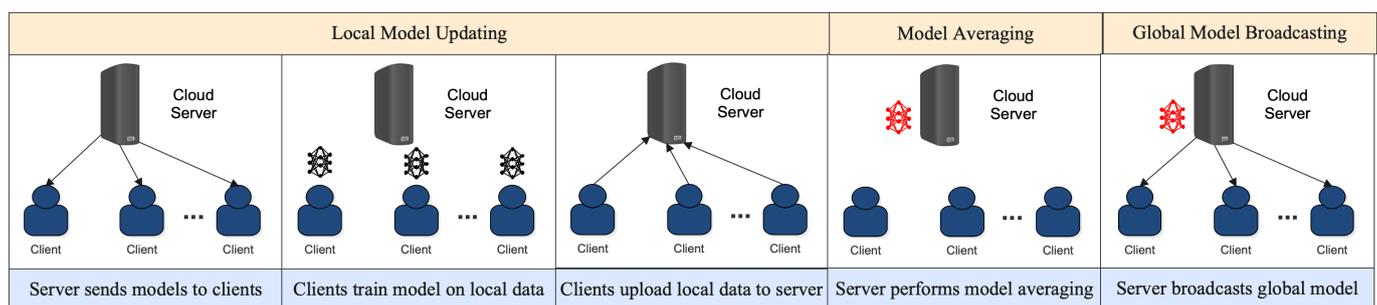


Figure 3. Workflow of communication protocol in FL.

3.1. Local Model Updating

Local model updating (LMU) is one of the key techniques used in FL to overcome communication deficiency [65]. In LMU, each participating device trains the shared model on its local data, and only the updated parameters are sent to the central server for aggregation [66,67]. This approach significantly reduces the amount of data that needs to be transmitted over the network, thereby reducing communication costs and latency.

However, several factors can affect the performance of LMU in FL, including the quality and quantity of local data, the frequency of updates, and the selection of participating

devices. Below, we discuss some of these factors and their impact on the communication efficiency of LMU in FL:

- **Quality and quantity of local data:** The quality and quantity of local data available on each participating device can significantly impact the performance of LMU in FL. If the local data are noisy or unrepresentative of the global dataset, it can lead to a poor model performance and increased communication costs [68,69]. Moreover, if the quantity of local data is too small, it can lead to overfitting and poor generalization, which can also affect the overall performance of the FL system [52,70]. Several techniques have been proposed to overcome these challenges, such as data filtering and data augmentation [71,72]. Data filtering involves removing noisy or irrelevant data from the local dataset before training the model. In contrast, data augmentation involves generating new data from the existing data to increase the quantity and diversity of the local dataset. These techniques can improve the quality and quantity of local data, thereby improving the performance of LMU in FL.
- **Frequency of updates:** The frequency of updates refers to how often the participating devices send their updated parameters to the central server for aggregation [73–75]. A higher frequency of updates can lead to faster convergence and an improved model performance but can also increase communication costs and latency. However, a lower frequency of updates can reduce communication costs but may result in slower convergence and suboptimal model performance. Several approaches have been proposed to balance these trade-offs, such as asynchronous updates and adaptive learning rates [76,77]. Asynchronous updates allow participating devices to update the shared model at their own pace, which can reduce communication cost and latency but may lead to slower convergence. Adaptive learning rates adjust the learning rate based on the frequency of updates, which can improve convergence and reduce communication costs.
- **Selection of participating devices:** The selection of participating devices in FL can significantly impact the performance of LMU [49,78]. If the participating devices are too few or diverse, it can lead to poor model generalization and increased communication costs. Moreover, if the participating devices are biased toward a particular subset of the data, it can lead to a poor model performance and increased communication costs. Several techniques have been proposed to overcome these challenges, such as stratified sampling [79] and weighted aggregation [80]. Stratified sampling involves selecting participating devices based on their similarity to the global dataset, which can improve model generalization and reduce communication costs. Weighted aggregation involves assigning different weights to the participating devices based on their local data quality and quantity, which can improve model performance and reduce communication costs.

3.2. Model Averaging

Model averaging is a popular technique used in FL to overcome the communication deficiency problem [81]. In particular, model averaging involves training multiple models on different devices and then combining the models to generate a final model that is more accurate than any individual model [82]. Below, we discuss the model averaging technique in detail and how it can help overcome communication deficiency in FL.

The model averaging technique involves training multiple models using the same training data on different devices. Each device trains its own model using its local data, and the models are then combined to generate a final model that is more accurate than any individual model [83,84]. The models are combined by taking the average of the weights of the individual models. This technique is known as “*Weighted Average Federated Learning*” [85].

Weighted Average FL works as follows. Let W_1, W_2, \dots, W_N be the weights of N individual models trained on different devices [86]. The final model is generated by taking

the weighted average of the weights of the individual models, where the weights are determined according to their accuracy. That is,

$$\text{Final Weight} = \frac{(w_1 \times W_1 + w_2 \times W_2 + \dots + w_N \times W_N)}{(w_1 + w_2 + \dots + w_N)}, \quad (1)$$

where w_1, w_2, \dots, w_N are the weights determined by the accuracy of individual models, and W_1, W_2, \dots, W_N are the weights of the corresponding models [87].

The weights of the individual models are determined based on their accuracy. Models that perform better on the local data are given higher weights, and models that perform poorly are given lower weights [88]. The weights are updated after each round of training, and the process is repeated until convergence.

The model averaging technique has several advantages over other techniques used in FL. First, it reduces the impact of communication deficiency by allowing each device to train its own model locally. This reduces the amount of communication required between the devices, which is particularly important in scenarios where the communication channel is limited. Second, it improves the accuracy of the final model by combining the knowledge of multiple models. This is particularly useful in scenarios where the local data are diverse and different devices have different data distributions.

In addition, the model averaging technique has been successfully used in several applications, including image classification, natural language processing, and recommendation systems [89]. For example, in image classification, multiple models are trained on different devices using different subsets of the training data [90]. The models are then combined using model averaging to generate a final model that is more accurate than any individual model. This technique has been shown to improve the accuracy of image classification models by up to 20%.

However, there are also some challenges associated with the model averaging technique [91]. One of the main challenges is the selection of the weights of the individual models. The weights should be selected in such a way that they reflect the accuracy of the models. If the weights are not selected correctly, the final model may not be accurate, and the performance may degrade. Another challenge is the convergence of the algorithm. Model averaging requires multiple training rounds, and the algorithm's convergence can be slow, particularly in scenarios where the local data are diverse [92].

3.3. Broadcasting the Global Model

Global model broadcasting is a crucial step in FL, where the locally trained models are aggregated to form a global model [93]. The global model represents the collective knowledge of all the edge devices and is used for making predictions and decisions. The global model must be communicated efficiently and effectively across all devices to achieve a high accuracy and high convergence rate [94]. However, this can be challenging in the presence of communication deficiency. In particular, the central server aggregates the model updates and computes the new global model, which is then broadcasted back to the edge devices [95,96]. There are two main approaches to global model broadcasting in FL: parameter-server-based and peer-to-peer.

In the parameter-server-based approach, a central server acts as a parameter server, which stores and manages the model parameters. The edge devices communicate with the parameter server to upload their local model updates and download the new global model [97]. The parameter server can update the global model by using a synchronous or asynchronous approach. In the synchronous approach, the edge devices upload their local model updates at regular intervals, and the parameter server updates the global model after receiving updates from all devices. In the asynchronous approach, the edge devices upload their local model updates as soon as they are ready, and the parameter server updates the global model in real time.

In the peer-to-peer approach, the edge devices communicate with each other directly to exchange their local model updates [98,99]. The devices can either use a fully connected topology or a decentralized topology to exchange their model updates. In a fully connected topology, each device communicates with all other devices to exchange their local model updates. In a decentralized topology, each device communicates with a subset of other devices to exchange their local model updates [33].

Communication deficiency is a major challenge in global model broadcasting in FL. The deficiency can be caused by a limited bandwidth, high latency, or network congestion [100,101]. The impact of communication deficiency can be severe, leading to slow convergence, a low accuracy, and even a divergence of the global model. In particular, a limited bandwidth can restrict the amount of data that can be transmitted between the edge devices and the central server. This can result in delayed model updates and slower convergence of the global model. High latency can also affect the performance of FL, leading to delayed model updates and the slower convergence of the global model. Network congestion can further exacerbate the problem, as it can cause packet loss and delay in model updates [102].

Several approaches have been proposed to mitigate communication deficiency in global model broadcasting in FL. Compression is one of the most effective approaches, where the model updates are compressed before transmission to reduce the data size [103]. Compression can significantly reduce the amount of data that needs to be transmitted, mitigating the impact of a limited bandwidth and network congestion. Another approach is network optimization techniques that can be used to improve communication efficiency between the edge devices and the central server [104,105]. This can be achieved through various methods, such as adaptive network scheduling, dynamic network reconfiguration, or traffic engineering. These techniques can help optimize the network resources and reduce the impact of network congestion and the latency impact. Model aggregation techniques can also be used to improve the efficiency of global model broadcasting. This can be achieved through various methods, such as federated averaging [106], decentralized optimization [107], or hierarchical aggregation [108]. These techniques can help to reduce the amount of data that needs to be transmitted and improve the convergence rate of the global model.

4. Resource Management

Managing resources is critical for the success of FL, which relies on a network of devices to train a machine learning model collaboratively [109]. In addition to computational and communication resources, the availability and quality of edge and server resources can significantly impact the performance of FL systems. In Table 4, we show the categorization of FL resources in terms of the edge and server. In addition, Figure 4 distinctly portrays the myriad techniques deployed for both client and server resource management in the context of federated learning. By effectively managing these resources, we can reduce communication costs and improve the efficiency and accuracy of FL models.

Table 4. Categorization of FL resources.

Resource	Edge Resource	Server Resource
Data Storage	Local Storage	Distributed Storage
Data Aggregation	Local Aggregation	Distributed Aggregation
Data Processing	Local Processing	Cloud Processing
Data Security	Local Encryption	Cloud Encryption

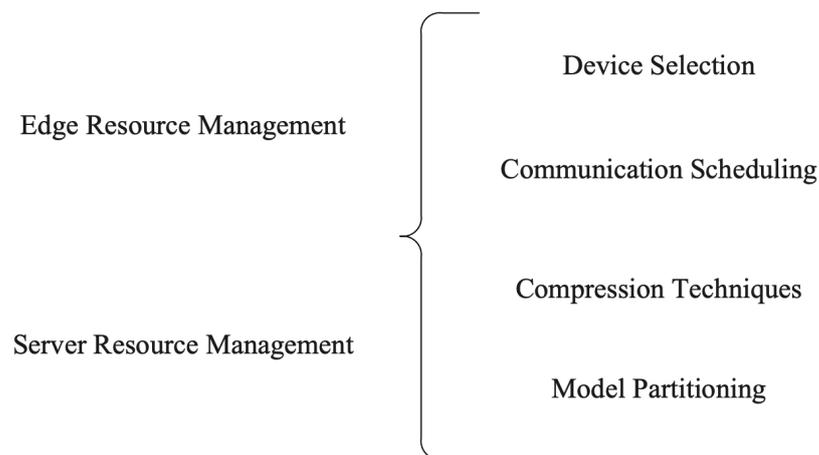


Figure 4. Techniques for clients and server resource management in FL.

4.1. Edge Resource Management

Edge resources refer to the computing and storage resources available on devices participating in the FL process. Edge devices typically have limited resources compared to cloud servers, which makes managing these resources a critical task in FL [110]. Effective edge resource management can help reduce communication costs and improve the overall performance of the FL system.

4.1.1. Device Selection

The first step in edge resource management is selecting appropriate devices for FL. Edge devices include smartphones, tablets, sensors, and other IoT devices. These devices vary in their processing power, memory capacity, battery life, and network connectivity. Therefore, selecting appropriate edge devices is critical for ensuring efficient resource management in FL [111].

One way to select edge devices is based on their processing power. Devices with more processing power can handle more complex machine learning models and computations [64]. However, devices with more processing power also tend to consume more energy, which can limit their battery life. Therefore, selecting devices with the right balance of processing power and energy efficiency is important. Another factor to consider when selecting edge devices is their memory capacity [112]. Devices with more memory can store more data and models, reducing the need for frequent communication with the central server. However, devices with limited memory can bottleneck in FL, especially when dealing with large datasets or models [113].

Network connectivity is another important factor to consider when selecting edge devices. Devices with reliable and high-speed network connectivity can communicate with the central server more efficiently, while devices with poor connectivity may experience delays or errors during communication [114,115]. In general, selecting appropriate edge devices depends on the specific use case and the requirements of the FL system. One common approach is to use a mix of devices with different characteristics to balance the trade-offs between processing power, memory, energy efficiency, and network connectivity.

4.1.2. Communication Scheduling

Communication scheduling is another important aspect of edge resource management in FL. Communication scheduling refers to exchanging data and models between edge devices and the central server [62,116]. Communication scheduling involves deciding when and how frequently to communicate and which devices to communicate with.

One strategy for communication scheduling is to schedule communication based on the availability and capacity of the edge devices. Devices with limited resources can be scheduled to communicate less frequently, while devices with more resources can be scheduled to communicate more frequently. This approach can help reduce the overall

communication costs of the FL system [117]. Another strategy for communication scheduling is to schedule communication based on the data and model updates [118]. Devices with more recent updates can be scheduled to communicate more frequently, while devices with older updates can be scheduled to communicate less frequently. This approach can help ensure the most relevant and up-to-date data and models are used in the FL process.

In addition, the communication schedule can also consider the network conditions and latency of the edge devices. Devices with poor network conditions or high latency can be scheduled to communicate during periods of low network traffic or when network conditions improve [119]. This approach can help reduce communication errors and delays in the FL system. Effective communication scheduling can help balance the trade-offs between communication costs and model accuracy and ensure the efficient use of edge resources.

4.1.3. Compression Techniques

Compression techniques are important for managing edge resources in FL. In particular, compression techniques involve reducing the data size and exchanging models between edge devices and the central server without sacrificing model accuracy [120].

The need for compression arises due to the limited resources available on edge devices. Edge devices typically have a limited storage capacity and network bandwidth, making transmitting large amounts of data and models challenging [121]. Compression techniques can help reduce the amount of data and models transmitted, making performing FL on edge devices with limited resources possible. There are several techniques for compressing data and models in FL. One common technique is quantization, which involves reducing the precision of the data and models [122]. For example, quantization can be used to represent the data and models as integers with a lower precision instead of transmitting floating-point numbers with a high precision. This can significantly reduce the size of the data and models transmitted without sacrificing much accuracy. Another technique for compressing data and models is pruning, which involves removing redundant or unnecessary parameters from the model [123–125]. Pruning can help reduce the model's size, making it easier to transmit over the network. However, pruning can also lead to a reduction in model accuracy if too many parameters are removed. Another technique for compressing data and models is knowledge distillation, which involves training a smaller model to mimic the behavior of a larger model [96,126]. The smaller model can then be used in place of the larger model, which can help reduce the model's size without sacrificing much accuracy. Knowledge distillation can be particularly effective when the larger model is complex and has many parameters.

In addition to these techniques, several compression algorithms are specifically designed for FL. For example, federated averaging (FedAvg) is a compression algorithm that involves averaging the model updates from multiple edge devices, which can help reduce the amount of data transmitted between devices [10]. Another algorithm, FedProx, involves adding a penalty term to the loss function to encourage edge devices to stay close to the global model [127]. This can help reduce the amount of data transmitted while maintaining model accuracy. By reducing the size of the data and models transmitted between edge devices and the central server, compression techniques can help reduce communication costs and improve the overall performance of the FL system.

4.1.4. Model Partitioning

Model partitioning is another critical component of FL systems, as it involves dividing the machine learning model into smaller submodels that can be trained on individual devices. Model partitioning aims to reduce the amount of communication required between devices while ensuring that the model's overall accuracy is not compromised [128].

Several strategies have been developed for model partitioning in FL systems. One common approach is vertical partitioning, where the model is divided based on the features or attributes being used [129]. For example, in an image recognition model, one device

may be responsible for training the feature extraction layer, while another device may train the classification layer. This approach can be particularly useful when the model has many features, allowing the devices to focus on a subset of the features [130]. Another approach is horizontal partitioning, where the model is divided based on the data being used [88,131]. For example, each device may train the model on a specific subset of the training data. This approach can be particularly useful when the data are distributed across multiple devices and transferring the entire dataset to a central server would be impractical. A third approach is hybrid partitioning, where a combination of vertical and horizontal partitioning is used to divide the model [132,133]. For example, the model may be partitioned vertically based on the features, and each feature may be further partitioned horizontally across multiple devices. However, the goal should always be to minimize the amount of communication required between devices while maintaining the model's overall accuracy.

4.2. Server Resource Management

Server resource management is a crucial aspect of FL that is responsible for optimizing the utilization of server resources to enhance the efficiency and accuracy of FL models [134,135]. A server's role in FL is coordinating and managing communication and computation among the participating edge devices. The server needs to allocate computational and communication resources optimally to ensure that the participating devices' requirements are met while minimizing the communication costs and enhancing the FL model's accuracy.

4.2.1. Device Selection

Device selection is a critical aspect of server resource management in FL. In an FL system, edge devices train a local model using their data and then communicate the model updates to the server [136,137]. The server aggregates the updates from all devices to create a global model. However, not all devices are suitable for participating in FL for several reasons, such as a low battery life, poor network connectivity, or low computation power. Therefore, the server must select the most suitable devices to participate in FL to optimize resource utilization and enhance model accuracy [138]. The device selection process can be based on several factors: the device computation power, network bandwidth, battery life, and data quality. A popular approach for device selection is to use a machine learning model that predicts the device's contribution to the global model [139]. The server can use the model's predictions to select the devices that are likely to provide the most significant contribution to the global model.

4.2.2. Communication Scheduling

The server needs to allocate communication resources optimally to ensure that the participating devices' updates are timely while minimizing communication costs. In FL, devices communicate with the server over wireless networks, which are prone to communication delays, packet losses, and network congestion [140]. Therefore, the server must effectively schedule communication between devices and the server. The communication schedule can be based on several factors, such as the device availability, network congestion, and data priority. A popular approach for communication scheduling is to use a priority-based scheduling algorithm that prioritizes the communication of high-priority data over low-priority data [141]. The server can use the device's data priority to schedule the communication effectively, which helps to reduce the communication delay and enhance the model accuracy.

4.2.3. Compression Techniques

In FL, the server receives updates from all participating devices, which can be significant in size. The size of the updates can be reduced by applying compression techniques to the updates before sending them to the server. Compression reduces the communi-

cation and the server's computational costs [142]. The compression techniques can be based on several factors, such as the update's sparsity, the update's structure, and the update's importance.

4.2.4. Model Partitioning

The model partitioning can be based on several factors, such as the model's size, the model's complexity, and the available server resources [143]. A popular model partitioning approach is the model distillation technique, which distills the global model into a smaller submodel [144]. The server can use the model distillation technique to partition the model into several submodels that can be trained and stored on different servers [145]. Another approach for model partitioning is to use the model parallelism technique, which splits the model into smaller parts that can be trained simultaneously on different servers [146,147]. The server can use the model parallelism technique to partition the model into smaller submodels that can be trained in parallel, significantly reducing the training time and improving the model accuracy.

5. Client Selection

The process of selecting appropriate clients for FL is a critical component of building successful FL systems. In this section, we will discuss various considerations that should be considered when selecting clients for FL, including factors such as device heterogeneity, device adaptability, incentive mechanisms, and adaptive aggregation. In Table 5, we show a comparison of each of those factors.

Table 5. Comparison of factors that can be considered for client selection in FL.

Device Heterogeneity	Device Adaptability	Incentive Mechanism	Adaptive Aggregation
Categorize devices	Assess device capability	Assign rewards	Aggregate according to device type
Evaluate device resources	Monitor device performance	Balance rewards	Adjust aggregation strategy
Consider device availability	Check device compatibility	Set rewards based on participation	Consider data privacy
Analyze device specifications	Identify device limitations	Assign rewards based on data quality	Adapt to changes in data distribution
Evaluate device trustworthiness	Assess device reliability	Offer rewards for data computation	Change aggregation frequency
Consider device latency	Determine device storage capacity	Provide rewards for data transmission	Monitor device performance
Check device battery level	Examine device memory usage	Create rewards for data accuracy	Adapt to changing device configurations

5.1. Device Heterogeneity

Device heterogeneity refers to the variety of devices and their characteristics that participate in an FL system. The heterogeneity of devices presents several challenges in FL, including system heterogeneity, statistical heterogeneity, and non-iid-ness [148].

5.1.1. System Heterogeneity

System heterogeneity refers to differences in the hardware, software, and networking capabilities of the devices participating in the FL system. The heterogeneity of these devices can lead to significant performance disparities and make it difficult to distribute and balance the workload among the devices [149]. These discrepancies can cause communication and synchronization issues, leading to slow convergence rates and increased communication costs. To address these issues, several techniques have been proposed, including device selection algorithms that select devices with similar hardware and software configura-

tions and adaptive communication schemes that adjust the communication frequency and message sizes based on the characteristics of the devices [150–152].

5.1.2. Statistical Heterogeneity

Statistical heterogeneity refers to the differences in the data distributions across the devices participating in the FL system. In an ideal FL system, the data should be identically and independently distributed (IID) across all devices, allowing the global model to be trained effectively [153,154]. However, in practice, the data are often non-IID, which can lead to a poor model performance. For example, if one device has significantly more data points for a specific class than others, the global model may become biased toward that class. Several techniques have been proposed to mitigate this issue, including data sampling [155], which involves selecting representative subsets of data from each device to achieve a more balanced distribution across devices, and data aggregation techniques that weigh the contribution of each device's update based on their local data distribution [156].

5.1.3. Non-IID-Ness

Non-iid-ness refers to the situation where the data distribution across the devices significantly differs from the global distribution. This is a common challenge in FL scenarios, where devices may collect data from different sources or have unique user behavior patterns [157]. The presence of non-iid-ness can lead to slower convergence rates and a poor model performance, as the global model may not accurately represent the data distribution across all devices [21,158]. To address non-iid-ness, several techniques have been proposed, including model personalization, which involves training personalized models for each device based on their local data distribution, and transfer learning, which involves leveraging knowledge learned from similar domains to improve model performance on non-iid data [159–161].

5.2. Device Adaptivity

Device adaptivity allows devices to adjust their participation in FL, which has emerged as an essential technique to reduce communication costs. Here, we will discuss two critical aspects of device adaptivity: flexible participation and partial updates.

5.2.1. Flexible Participation

Flexible participation allows devices to determine the extent of their involvement in FL based on their capabilities and resources. It allows devices to choose how much data they will contribute, how many communication rounds they will participate in, and when they will participate [162,163]. Flexible participation can significantly reduce communication costs by enabling devices with limited resources to participate in FL without overburdening their systems.

One way to achieve flexible participation is to use dynamic client selection. Dynamic client selection involves selecting clients based on their data quality, availability, and computation capabilities [164]. This approach can significantly reduce communication costs by only selecting a subset of clients to participate in each round of training. Another approach to achieving flexible participation is to use selective transfer learning, where models are selectively transferred from high-capability devices to low-capability devices to minimize communication costs. This approach is particularly effective when training large models with limited resources [165].

5.2.2. Partial Updates

Partial updates allow devices to transmit only a portion of their model updates to the central server instead of transmitting the entire update [166]. This approach can significantly reduce communication costs by reducing the amount of data transmitted between devices. Partial updates can be achieved in several ways, including compressing

the model updates, using differential privacy to obscure the update, and using gradient sparsification to reduce the update's size [167].

Compression techniques, such as quantization, pruning, and sparsification, can be used to reduce the size of the model updates [8]. Quantization involves reducing the precision of the model parameters to reduce their size. Pruning involves removing redundant or insignificant parameters from the model. Sparsification involves setting some parameters to zero to reduce the size of the model update. Differential privacy can be used to obscure the model update by adding random noise to the update [168]. Gradient sparsification can reduce the update's size by only transmitting the most significant gradient values.

5.3. Incentive Mechanism

One of the main challenges in minimizing communication costs in FL is incentivizing the clients to cooperate and share their local model updates with the central server. Incentives can encourage clients to participate actively and contribute to the system, leading to a better performance and scalability [97,169,170]. However, designing effective incentive mechanisms is not straightforward and requires careful consideration of various factors. Figure 5 provides a detailed visualization of the FL incentive mechanism. It offers insights into how different stakeholders, from data providers to model trainers, are motivated to participate in the federated ecosystem, ensuring that contributions are recognized and rewarded appropriately, fostering a collaborative and sustainable environment.

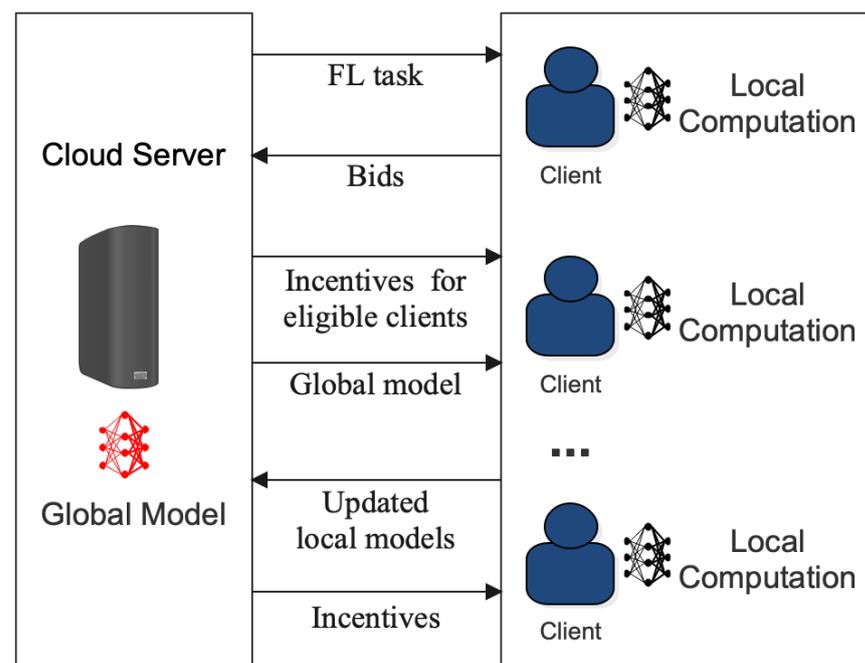


Figure 5. Process of incentive mechanism in FL.

Different types of incentive mechanisms can be used to encourage participation in FL. Some of the commonly used incentive mechanisms are explained below:

- **Monetary incentives:** Monetary incentives involve rewarding the clients with a monetary value for their contributions. This approach can effectively motivate the clients to contribute actively to the system [171]. However, it may not be practical in all situations, as it requires a budget to support the incentive program.
- **Reputation-based incentives:** Reputation-based incentives are based on the principle of recognition and reputation. The clients who contribute actively and provide high-quality updates to the system can be recognized and rewarded with a higher reputation score [172]. This approach can effectively motivate the clients to contribute to the system actively.

- **Token-based incentives:** Token-based incentives involve rewarding the clients with tokens that can be used to access additional features or services [173]. This approach can effectively motivate the clients to contribute actively to the system and help build a vibrant ecosystem around the FL system.

The choice of incentive mechanism depends on the system's specific requirements and the clients' nature. In general, the incentive mechanism should be designed to align the clients' interests with the system's goals. One of the critical factors to consider while designing an incentive mechanism for communication costs in FL is the clients' privacy concerns [174]. In FL, the clients' data are typically stored locally on their devices, and only the model updates are shared with the central server. Therefore, the incentive mechanism should not compromise the privacy of the client's data.

Various privacy-preserving techniques can be used to address the clients' privacy concerns. For example, differential privacy can be used to ensure that the model updates do not reveal any sensitive information about the client's data [175]. In this approach, noise is added to the model updates before sharing them with the central server, making extracting any individual information from the updates difficult. Another critical factor to consider while designing an incentive mechanism is the system's fairness [176]. The incentive mechanism should be designed to ensure that all the clients are treated fairly and that their contributions are appropriately recognized. Fairness can be ensured by designing an incentive mechanism to reward the clients based on their contributions rather than their status or position in the system. Another critical aspect to consider while designing the incentive mechanism is the central server's level of control over the clients [177]. The incentive mechanism should be designed to ensure that the clients have a certain level of autonomy and control over their data. The clients should be free to decide whether to participate in the system or not, and their contributions should be voluntary.

5.4. Adaptive Aggregation

Adaptive aggregation is a method for reducing communication costs in FL systems. In FL, data are typically distributed across multiple devices, and the goal is to train a machine learning model using this decentralized data. To accomplish this, the data are typically aggregated on a central server, which can be computationally expensive and lead to high communication costs [178,179]. Adaptive aggregation seeks to mitigate these costs by dynamically adjusting the amount of aggregated data based on the communication bandwidth of the selected client [180].

The basic idea behind adaptive aggregation is to adjust the amount of data sent to the central server based on the available bandwidth of the devices. This means that devices with slow or limited connectivity can send fewer data, while faster or more reliable connectivity can send more data. Adaptive aggregation can reduce the overall communication costs of FL systems by adapting the amount of data sent [181].

There are several ways that adaptive aggregation can be implemented in FL systems. One approach is to use a threshold-based method, where each device sends a fixed amount of data until its bandwidth is exceeded, at which point it stops sending data [182]. This approach is simple and easy to implement. Still, it may not be very effective at reducing communication costs since it does not consider the variability of communication bandwidth across devices. A more sophisticated approach is a feedback-based method, where the amount of data sent by each device is adjusted based on feedback from the central server [183]. This feedback can be in the form of acknowledgments or error messages, which indicate whether the data received by the server were sufficient to update the model. Devices with faster or more reliable connectivity can send more data, while devices with slower or less reliable connectivity can be limited to sending smaller amounts of data. This approach can be more effective at reducing communication costs since it can adapt to the variability of communication bandwidth across devices. Another approach to adaptive aggregation is to use a learning-based method, where the amount of data sent by each device is adjusted based on past performance [184]. This can be performed using machine

learning techniques like reinforcement learning or neural networks. The system can learn to predict the optimal amount of data to send based on the communication bandwidth of the devices and adjust the amount of data sent accordingly. This approach can effectively reduce communication costs since it can adapt to the specific characteristics of the devices in the FL system.

One of the challenges of adaptive aggregation is determining the appropriate amount of data to send for each device. If too few data are sent, the model may not converge to an accurate solution, while if too many data are sent, the communication costs may be excessive [185]. This trade-off can be addressed by using techniques such as cross-validation, which can estimate the model's performance based on a subset of the data [88]. Another challenge is ensuring that the model is updated in a timely manner despite the variability in communication bandwidth across devices [186]. This can be addressed using techniques such as asynchronous updates, allowing devices to update the model independently and asynchronously [187].

6. Optimization Techniques

This section will discuss two key optimization techniques commonly used in FL: compression schemes and structured updates. Table 6 shows the pros and cons of those techniques.

Table 6. Pros and cons of optimization techniques in FL.

Technique	Pros	Cons
Compression Schemes		
Quantization	Reduced communication	Information loss
Sparsification	Lower bandwidth usage	Increased computation
Low-rank factorization	Efficient storage	Complexity in updating
Structured Updates		
Gradient sparsification	Reduced communication	Limited expressiveness
Weight differencing	Low memory requirement	Sensitivity to noise

6.1. Compression Schemes

Compression schemes involve techniques that reduce the models' size and gradients exchanged between the client devices and the central server. This is necessary because the communication costs of exchanging large models and gradients can be prohibitively high, especially when client devices have limited bandwidth or computing resources [30,188]. Various compression schemes can be used to address this issue, including quantization, sparsification, and low-rank factorization.

6.1.1. Quantization

Quantization is a popular technique that involves representing the model or gradient values using a smaller number of bits than their original precision [189]. For instance, instead of representing a model parameter using a 32 bit floating-point number, it can be represented using an 8 bit integer. This reduces the number of bits that need to be transmitted and can significantly reduce communication costs. However, quantization also introduces some errors in the model or gradient values, which can affect the quality of the learning process.

6.1.2. Sparsification

Sparsification is another commonly used compression technique that involves setting a large proportion of the model or gradient values to zero [190]. This reduces the number of non-zero values that need to be transmitted, which can result in significant communication savings. Sparsification can be achieved using techniques such as thresholding, random pruning, and structured pruning. However, sparsification can also introduce some errors

in the model or gradient values, which can impact the accuracy of the learning process. Some sparsification techniques are described below:

- **Thresholding** is a popular technique for sparsification that involves setting all model or gradient values below a certain threshold to zero [191]. This reduces the number of non-zero values that need to be transmitted, which can result in significant communication savings. The threshold can be set using various techniques, such as absolute thresholding, percentage thresholding, and dynamic thresholding. Absolute thresholding involves setting a fixed threshold for all values, whereas percentage thresholding involves setting a threshold based on the percentage of non-zero values. Dynamic thresholding involves adjusting the threshold based on the distribution of the model or gradient values [192].
- **Random pruning** is another sparsification technique that randomly sets some model or gradient values to zero [123]. This reduces the number of non-zero values that need to be transmitted and can result in significant communication savings. Random pruning can be achieved using techniques like Bernoulli sampling and stochastic rounding [193]. Bernoulli sampling involves setting each value to zero with a certain probability, whereas stochastic rounding involves rounding each value to zero with a certain probability.
- **Structured pruning** is a sparsification technique that sets entire rows, columns, or blocks of the model or gradient matrices to zero [194]. This reduces the number of non-zero values that need to be transmitted and can result in significant communication savings. Structured pruning can be achieved using various techniques like channel, filter, and tensor pruning. Channel pruning involves setting entire channels of the model to zero, whereas filter pruning involves setting entire model filters to zero. Tensor pruning involves setting entire blocks of the model to zero, which can be useful when the model has a structured block-wise pattern. Structured pruning can preserve the underlying structure of the model and can result in higher compression rates than random pruning [195]. Still, it may require more complex implementation and may introduce more errors in the model or gradient values.

6.1.3. Low-Rank Factorization

Low-rank factorization is a compression technique that involves representing the model or gradient matrices using a low-rank approximation [196,197]. This reduces the number of parameters that need to be transmitted and can significantly reduce communication costs. Low-rank factorization can be achieved using techniques such as Singular Value Decomposition (SVD) [198] and Principal Component Analysis (PCA) [199]. However, low-rank factorization can also introduce some errors in the model or gradient values, which can affect the quality of the learning process. The techniques are described below:

- **Singular Value Decomposition (SVD)**: SVD is a matrix factorization technique that decomposes a matrix X into three matrices A , B , and C such that $X = ABC^T$. Here, A and C are orthogonal matrices, and B is a diagonal matrix containing the singular values of X . The script T represents the transpose operator, which flips the rows and columns of a matrix. The singular values represent the amount of variation captured by each singular vector. By retaining only the $top - k$ singular values and their corresponding singular vectors, we can approximate the original matrix X with a lower rank matrix $X_k = A_k B_k C_k^T$, where A_k and C_k are the truncated orthogonal matrices, and B_k contains only the $top - k$ singular values [200].
- **Principal Component Analysis (PCA)**: PCA is a dimensionality reduction technique that can be used to compress data. Given a data matrix X , PCA aims to find a lower-dimensional representation of X that retains the maximum amount of variance. This is achieved by computing the eigenvectors of the covariance matrix of X and selecting the $top - k$ eigenvectors corresponding to the largest eigenvalues. The selected eigenvectors form a new orthogonal basis for the data, and the projection of X onto this basis yields the lower-dimensional representation of X [201].

6.2. Structured Updates

Structured updates are another important optimization technique in FL that can reduce communication costs by transmitting only the updates to the changed model parameters. This is necessary because, in many FL scenarios, only a small proportion of the client devices update their local models in each round of communication, and transmitting the entire model can be wasteful [11,202]. Structured updates involve identifying the parts of the model that have been updated and transmitting only those parts to the central server. Various techniques can be used to achieve structured updates, such as gradient sparsification and weight differencing [8].

6.2.1. Gradient Sparsification

Gradient sparsification is a technique used to reduce communication costs in FL. In this technique, only the important gradient values are sent instead of sending the complete gradient information [203]. This can be performed by setting a threshold value and sending only those gradients whose absolute value exceeds the threshold. This threshold can be adjusted depending on the compression and the model's performance [204]. By reducing the number of gradients sent, the communication costs can be significantly reduced while maintaining the model's accuracy.

6.2.2. Weight Differencing

Weight differencing is a technique used to reduce communication costs in FL. In this technique, only the differences between the current and previous model parameters are sent instead of sending the entire model parameters [205]. This can be performed by computing the difference between the model parameters at the end of each round and sending only the difference information. This technique reduces the amount of information sent over the network and thus reduces communication costs. However, it requires additional computation at each client to compute the difference and may not be suitable for all scenarios.

7. Future Directions

Despite the potential benefits, existing research on FL discusses several challenges associated with communication efficiency. Overcoming those challenges is crucial for harnessing the full potential of FL and realizing its benefits across diverse domains and applications. In Table 7, we briefly summarize the existing research challenges. In addition, below, we explore some possible future directions in FL to reduce communication costs. By leveraging the following techniques, we can improve the efficiency and scalability of FL algorithms and enable the training of machine learning models on increasingly large and diverse datasets.

Table 7. Summary of existing research challenges in FL related to communication efficiency.

Research Challenge	Brief Description
High Communication Overhead	FL requires transferring large amounts of data, which can lead to high communication costs.
Data Heterogeneity	Differences in data distribution across devices can affect model performance and require efficient communication strategies.
Latency	Variations in network conditions and device capabilities can cause latency issues, requiring efficient communication solutions.
Bandwidth Limitations	A limited bandwidth can cause slow model training and update propagation. The efficient use of the available bandwidth is a challenge.
Stragglers	Some devices may be slow to compute updates or fail to send updates, slowing down the learning process. The efficient handling of stragglers can improve communication efficiency.

Table 7. *Cont.*

Research Challenge	Brief Description
Scalability	As the number of participating devices increases, efficiently managing communications becomes more challenging.
Security	Efficiently ensuring secure and privacy-preserving communication is a significant challenge.
Device Failures	Devices may fail or drop out during the learning process, requiring robust communication protocols to handle these situations.
Resource Constraints	Devices participating in FL may have different computational resources, which can create challenges for efficient communication.
Data Synchronization	Ensuring all devices have the latest model updates for efficient learning can be a challenge, especially given the asynchronous nature of FL.
Noise in Gradients	Due to the decentralized nature of FL, there can be a high level of noise in the gradient updates, affecting the overall communication efficiency.
Compressed Communication	Due to bandwidth limitations, it may be necessary to compress data during transmission, which can lead to a loss of information and affect the learning process.

7.1. Edge Intelligence

Edge intelligence is a concept where machine learning models are deployed on the edge devices, such as smartphones, IoT devices, and sensors. By deploying the models on these devices, the communication costs are significantly reduced, as the data does not need to be transmitted to a central server for processing. Instead, the models can be trained locally on the edge devices, and only the model updates need to be communicated to the central server [206,207].

7.2. Quantum Computing

Quantum computing has the potential to revolutionize FL by enabling faster and more efficient computations [208]. Quantum computers can perform certain tasks currently infeasible with classical computers, such as factoring large numbers and solving optimization problems. This could lead to significant improvements in the efficiency of FL algorithms, which rely heavily on optimization.

7.3. Federated Transfer Learning

Federated transfer learning is a technique where models trained on one device or node can be transferred to another device or node, where they can be fine-tuned on local data. This approach can significantly reduce communication costs, as only the model updates are required to be communicated between the devices rather than the entire model [209].

7.4. Multi-Task Learning

Multi-task learning is a technique where a single model is trained on multiple related tasks simultaneously [210]. In FL, this approach can reduce communication costs by allowing the nodes to share their local models, which can be fine-tuned on other related tasks.

7.5. Federated Reinforcement Learning

Federated reinforcement learning is a technique where agents learn from their interactions with the environment, and the models are trained in a decentralized manner [211]. This approach can significantly reduce communication costs, as the agents can learn from their local experiences and only communicate the model updates to a central server.

7.6. Federated Meta-Learning

Federated meta-learning is a technique where a meta-model is trained on the local models of each node, and the meta-model is used to coordinate the training process [212]. This approach can reduce communication costs by allowing the nodes to share their local models, which can be used to improve the performance of the meta-model.

7.7. Hybrid Approaches

Hybrid approaches combine multiple techniques to achieve a better performance and reduce communication costs [213]. For example, a hybrid approach could combine edge intelligence with federated transfer learning, where models are trained on edge devices and transferred to a central server for fine-tuning.

7.8. Automatic Model Compression

Automatic model compression is a technique where machine learning models are compressed to reduce their size and complexity, which can significantly reduce communication costs [214]. This technique can be used with other approaches, such as federated transfer learning, to further reduce communication costs.

7.9. Federated Learning in Medical Fields

As federated learning (FL) continues its integration with the burgeoning realm of the Internet of Medical Things (IoMT), buttressed by the advanced capabilities of 6G, new horizons in healthcare appear imminent. The study [215] offers a glimpse into this future, showcasing 6G-enhanced FL in IoMT. Emerging challenges and opportunities in this confluence include enhancing real-time health monitoring and diagnostics while ensuring robust data privacy. The next frontier likely involves crafting tailored communication-efficient techniques that can accommodate the unique demands of medical diagnosis and treatment. There is a palpable anticipation for a healthcare paradigm where FL and 6G seamlessly intertwine, catalyzing more personalized, timely, and secure patient care [216]. Future endeavors in this domain will undoubtedly focus on harnessing these synergies for optimal healthcare outcomes.

8. Discussion and Analysis

While this survey has comprehensively detailed techniques addressing communication efficiency in FL, it is paramount to understand their inherent challenges, complexities, and potential benefits.

8.1. Challenges and Complexities

The juxtaposition of local model updating, model averaging, and broadcasting the global model hints at a delicate balance: optimizing one aspect can inadvertently impact another, leading to unforeseen communication bottlenecks.

Resource management, especially on the edge versus server-side, is not a straightforward binary. Factors like unpredictable client availability, diverse resource capabilities, and fluctuating network conditions make universal solutions elusive. The prominence given to client selection is noteworthy; yet, the task is not trivial. Deciding on “the most appropriate devices” involves not just current resource metrics but predictive insights into their future states, adding another layer of complexity.

Moreover, while optimization techniques like compression schemes and structured updates promise reduced communication costs, they come with their caveats. For instance, aggressive model compression might reduce data transfer but could also lead to degraded model accuracy. Structured updates, although efficient, may not always align with the non-i.i.d data distributions often seen in FL setups.

In light of the future directions presented, it is clear that while advancements such as quantum computing and federated meta-learning offer exciting prospects their practical

application in FL will introduce new challenges. It is imperative that, as we move forward, we are not just devising solutions but anticipating the trade-offs they bring to the fore.

8.2. Benefits of Energy-Efficient FL

Energy efficiency has recently emerged as an indispensable pillar in the realm of communication systems, predominantly due to the burgeoning demand for connected devices and the skyrocketing data exchange volumes. FL, with its inherent design, lends itself beneficially to this scenario. Some of its benefits are listed below:

- **Reduced data transmission:** At its core, FL minimizes the need for data centralization. Instead of transmitting extensive datasets, devices share compressed model updates. This direct reduction in data transmission not only conserves bandwidth but also considerably reduces the energy expended in the communication process, given that data transmission and reception are among the most energy-intensive operations in wireless communication.
- **Decentralized computation:** In FL, computations are performed at the edge, on user devices themselves. This decentralization aids in leveraging the collective computational prowess of these devices, reducing the burden on centralized servers. Consequently, servers consume less energy for computations, ensuring a more balanced and energy-efficient system.
- **Intelligent client participation:** Energy efficiency in FL is not just about reducing communication. It extends to judiciously determining which clients participate in the training. By selecting devices that are currently charging or have high battery levels, FL processes can minimize battery drain issues, leading to a more sustainable execution of federated tasks.
- **Adaptive communication protocols:** Modern FL implementations have started employing adaptive communication techniques. By assessing the network's current state, these techniques modulate the frequency and size of model updates. Such dynamism ensures that devices communicate optimally, preserving energy in low-bandwidth or unstable network conditions.
- **Synergy with modern hardware:** With the advent of energy-efficient hardware tailored for AI and ML tasks, FL can further amplify energy savings. By integrating with low-power neural network accelerators, for instance, the computational aspect of FL becomes even more energy efficient.

While energy efficiency introduces undeniable advantages, it is paramount to integrate it thoughtfully into the FL paradigm. The challenge is ensuring that the pursuit of energy savings does not compromise the model's accuracy or the system's responsiveness.

9. Conclusions

This survey paper has thoroughly analyzed the limitations and future aspects of communication costs in federated learning. We have explored the fundamentals of federated learning, the challenges associated with communication deficiency, resource management, client selection, and optimization techniques. The survey has highlighted the need to address communication costs to improve the efficiency and scalability of federated learning. The future directions of federated learning with respect to communication costs have also been identified. This survey paper provides a valuable resource for researchers and practitioners working on federated learning and inspires further research in this area.

Author Contributions: All authors contributed equally to the work. All authors have read and agreed to the published version of the manuscript.

Funding: These research results were obtained from research commissioned by the National Institute of Information and Communications Technology (NICT), JAPAN.

Data Availability Statement: The original contributions presented in the study and included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this research article.

References

1. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
2. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef]
3. AbdulRahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J.* **2020**, *8*, 5476–5497. [CrossRef]
4. Wang, T.; Rausch, J.; Zhang, C.; Jia, R.; Song, D. A principled approach to data valuation for federated learning. In *Federated Learning: Privacy and Incentive*; Springer: Cham, Switzerland, 2020; pp. 153–167.
5. Kaiwartya, O.; Kaushik, K.; Gupta, S.K.; Mishra, A.; Kumar, M. *Security and Privacy in Cyberspace*; Springer Nature: Singapore, 2022.
6. Luo, B.; Li, X.; Wang, S.; Huang, J.; Tassiulas, L. Cost-effective federated learning design. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021; pp. 1–10. Available online: <https://ieeexplore.ieee.org/document/9488679> (accessed on 19 August 2023).
7. Shahid, O.; Pouriyeh, S.; Parizi, R.M.; Sheng, Q.Z.; Srivastava, G.; Zhao, L. Communication efficiency in federated learning: Achievements and challenges. *arXiv* **2021**, arXiv:2107.10996.
8. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
9. Tran, N.H.; Bao, W.; Zomaya, A.; Nguyen, M.N.; Hong, C.S. Federated learning over wireless networks: Optimization model design and analysis. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, 2019; pp. 1387–1395. Available online: <https://ieeexplore.ieee.org/document/8737464> (accessed on 19 August 2023).
10. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, 2017; pp. 1273–1282. Available online: <https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf> (accessed on 19 August 2023).
11. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, B.; et al. Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **2019**, *1*, 374–388.
12. Xu, J.; Du, W.; Jin, Y.; He, W.; Cheng, R. Ternary compression for communication-efficient federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1162–1176. [CrossRef]
13. Reiszadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2020; pp. 2021–2031. Available online: <http://proceedings.mlr.press/v108/reiszadeh20a/reiszadeh20a.pdf> (accessed on 19 August 2023).
14. Lorincz, J.; Klarin, Z.; Begusic, D. Advances in Improving Energy Efficiency of Fiber–Wireless Access Networks: A Comprehensive Overview. *Sensors* **2023**, *23*, 2239. [CrossRef]
15. Lorincz, J.; Klarin, Z. How trend of increasing data volume affects the energy efficiency of 5g networks. *Sensors* **2021**, *22*, 255. [CrossRef]
16. Al-Abiad, M.S.; Obeed, M.; Hossain, M.; Chaaban, A. Decentralized aggregation for energy-efficient federated learning via D2D communications. *IEEE Trans. Commun.* **2023**, *71*, 3333–3351. [CrossRef]
17. Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M.V.; Herrera, F.; Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Inf. Fusion* **2023**, *90*, 148–173. [CrossRef]
18. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]
19. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
20. Xia, Q.; Ye, W.; Tao, Z.; Wu, J.; Li, Q. A survey of federated learning for edge computing: Research problems and solutions. *High-Confid. Comput.* **2021**, *1*, 100008. [CrossRef]
21. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated learning on non-IID data: A survey. *Neurocomputing* **2021**, *465*, 371–390. [CrossRef]
22. Nguyen, J.; Malik, K.; Zhan, H.; Yousefpour, A.; Rabbat, M.; Malek, M.; Huba, D. Federated learning with buffered asynchronous aggregation. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2022; pp. 3581–3607. Available online: <https://proceedings.mlr.press/v151/nguyen22b/nguyen22b.pdf> (accessed on 19 August 2023).
23. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.* **2023**, *55*, 1–31. [CrossRef]
24. Ghimire, B.; Rawat, D.B. Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. *IEEE Internet Things J.* **2022**, *9*, 8229–8249. [CrossRef]

25. Gupta, R.; Alam, T. Survey on federated-learning approaches in distributed environment. *Wirel. Pers. Commun.* **2022**, *125*, 1631–1652. [[CrossRef](#)]
26. Boobalan, P.; Ramu, S.P.; Pham, Q.V.; Dev, K.; Pandya, S.; Maddikunta, P.K.R.; Gadekallu, T.R.; Huynh-The, T. Fusion of federated learning and industrial Internet of Things: A survey. *Comput. Netw.* **2022**, *212*, 109048. [[CrossRef](#)]
27. Al-Quraan, M.; Mohjazi, L.; Bariah, L.; Centeno, A.; Zoha, A.; Arshad, K.; Assaleh, K.; Muhaidat, S.; Debbah, M.; Imran, M.A. Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 957–979. [[CrossRef](#)]
28. Zhao, Z.; Mao, Y.; Liu, Y.; Song, L.; Ouyang, Y.; Chen, X.; Ding, W. Towards Efficient Communications in Federated Learning: A Contemporary Survey. *J. Frankl. Inst.* **2023**, *360*, 8669–8703. [[CrossRef](#)]
29. Sikandar, H.S.; Waheed, H.; Tahir, S.; Malik, S.U.; Rafique, W. A Detailed Survey on Federated Learning Attacks and Defenses. *Electronics* **2023**, *12*, 260. [[CrossRef](#)]
30. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [[CrossRef](#)]
31. Wang, Z.; Nakazato, J.; Asad, M.; Javanmardi, E.; Tsukada, M. Overcoming Environmental Challenges in CAVs through MEC-based Federated Learning. In Proceedings of the 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), 2023; pp. 151–156. Available online: <https://ieeexplore.ieee.org/document/10200688> (accessed on 19 August 2023).
32. Kulkarni, V.; Kulkarni, M.; Pant, A. Survey of personalization techniques for federated learning. In Proceedings of the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020; pp. 794–797. Available online: <https://ieeexplore.ieee.org/document/9210355> (accessed on 19 August 2023).
33. Roy, A.G.; Siddiqui, S.; Pölsterl, S.; Navab, N.; Wachinger, C. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv* **2019**, arXiv:1905.06731.
34. Li, W.; Chen, J.; Wang, Z.; Shen, Z.; Ma, C.; Cui, X. Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, early access.
35. Hegedűs, I.; Danner, G.; Jelasity, M. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *J. Parallel Distrib. Comput.* **2021**, *148*, 109–124. [[CrossRef](#)]
36. Kang, J.; Xiong, Z.; Niyato, D.; Zou, Y.; Zhang, Y.; Guizani, M. Reliable federated learning for mobile networks. *IEEE Wirel. Commun.* **2020**, *27*, 72–80. [[CrossRef](#)]
37. Ye, Y.; Li, S.; Liu, F.; Tang, Y.; Hu, W. EdgeFed: Optimized federated learning based on edge computing. *IEEE Access* **2020**, *8*, 209191–209198. [[CrossRef](#)]
38. Yao, X.; Huang, C.; Sun, L. Two-stream federated learning: Reduce the communication costs. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), 2018; pp. 1–4. Available online: <https://ieeexplore.ieee.org/document/8698609> (accessed on 19 August 2023).
39. Ye, D.; Yu, R.; Pan, M.; Han, Z. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access* **2020**, *8*, 23920–23935. [[CrossRef](#)]
40. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust aggregation for federated learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [[CrossRef](#)]
41. Ma, X.; Zhang, J.; Guo, S.; Xu, W. Layer-wised model aggregation for personalized federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 10092–10101. Available online: https://openaccess.thecvf.com/content/CVPR2022/html/Ma_Layer-Wised_Model_Aggregation_for_Personalized_Federated_Learning_CVPR_2022_paper.html (accessed on 19 August 2023).
42. Hu, L.; Yan, H.; Li, L.; Pan, Z.; Liu, X.; Zhang, Z. MHAT: An efficient model-heterogenous aggregation training scheme for federated learning. *Inf. Sci.* **2021**, *560*, 493–503. [[CrossRef](#)]
43. Deng, Y.; Lyu, F.; Ren, J.; Chen, Y.C.; Yang, P.; Zhou, Y.; Zhang, Y. Fair: Quality-aware federated learning with precise user incentive and model aggregation. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021; pp. 1–10. Available online: <https://ieeexplore.ieee.org/document/9488743> (accessed on 19 August 2023).
44. Xu, R.; Baracaldo, N.; Zhou, Y.; Anwar, A.; Ludwig, H. Hybridalpha: An efficient approach for privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019; pp. 13–23. Available online: https://dl.acm.org/doi/abs/10.1145/3338501.3357371?casa_token=npneF7k5jXMAAAAA:16iC0bT3mCxKmPch0GrVIR_qlO72nQKPvwx6zICPYhHreVHWMaDKJEiv9dGEn9NTC7YSHDY6J5MDXg (accessed on 19 August 2023).
45. Gu, B.; Xu, A.; Huo, Z.; Deng, C.; Huang, H. Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6103–6115. [[CrossRef](#)] [[PubMed](#)]
46. Alam, T.; Gupta, R. Federated learning and its role in the privacy preservation of IoT devices. *Future Internet* **2022**, *14*, 246. [[CrossRef](#)]
47. Chen, M.; Shlezinger, N.; Poor, H.V.; Eldar, Y.C.; Cui, S. Communication-efficient federated learning. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2024789118. [[CrossRef](#)] [[PubMed](#)]
48. Asad, M.; Moustafa, A.; Ito, T.; Aslam, M. Evaluating the communication efficiency in federated learning algorithms. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2021; pp. 552–557. Available online: <https://ieeexplore.ieee.org/document/9437738> (accessed on 19 August 2023).

49. Zhang, W.; Wang, X.; Zhou, P.; Wu, W.; Zhang, X. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access* **2021**, *9*, 24462–24474. [CrossRef]
50. Albelaihi, R.; Yu, L.; Craft, W.D.; Sun, X.; Wang, C.; Gazda, R. Green Federated Learning via Energy-Aware Client Selection. In Proceedings of the GLOBECOM 2022-2022 IEEE Global Communications Conference, 2022; pp. 13–18. Available online: <https://ieeexplore.ieee.org/document/10001569> (accessed on 19 August 2023).
51. Asad, M.; Moustafa, A.; Rabhi, F.A.; Aslam, M. THF: 3-way hierarchical framework for efficient client selection and resource management in federated learning. *IEEE Internet Things J.* **2021**, *9*, 11085–11097. [CrossRef]
52. Chai, Z.; Ali, A.; Zawad, S.; Truex, S.; Anwar, A.; Baracaldo, N.; Zhou, Y.; Ludwig, H.; Yan, F.; Cheng, Y. Tifl: A tier-based federated learning system. In Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, 2020; pp. 125–136. Available online: https://dl.acm.org/doi/abs/10.1145/3369583.3392686?casa_token=H-rLbMWgQcgAAAAA:4W7rio6RI5d19VplBX6jmf7vCoxYDmQzQSFoeliE75eG7aQZcvBGvs5v8Sdy1SiEISKPdmjAcqzx5Q (accessed on 19 August 2023).
53. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Netw.* **2019**, *33*, 156–165. [CrossRef]
54. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5098–5107. [CrossRef]
55. Tianqing, Z.; Zhou, W.; Ye, D.; Cheng, Z.; Li, J. Resource allocation in IoT edge computing via concurrent federated reinforcement learning. *IEEE Internet Things J.* **2021**, *9*, 1414–1426. [CrossRef]
56. Kang, J.; Li, X.; Nie, J.; Liu, Y.; Xu, M.; Xiong, Z.; Niyato, D.; Yan, Q. Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2966–2977. [CrossRef]
57. Sun, P.; Che, H.; Wang, Z.; Wang, Y.; Wang, T.; Wu, L.; Shao, H. Pain-FL: Personalized privacy-preserving incentive for federated learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3805–3820. [CrossRef]
58. Li, Y.; Tao, X.; Zhang, X.; Liu, J.; Xu, J. Privacy-preserved federated learning for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8423–8434. [CrossRef]
59. Zeng, T.; Semiari, O.; Chen, M.; Saad, W.; Bennis, M. Federated learning on the road autonomous controller design for connected and autonomous vehicles. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 10407–10423. [CrossRef]
60. Ng, J.S.; Lim, W.Y.B.; Xiong, Z.; Cao, X.; Niyato, D.; Leung, C.; Kim, D.I. A hierarchical incentive design toward motivating participation in coded federated learning. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 359–375. [CrossRef]
61. Liu, L.; Zhang, J.; Song, S.; Letaief, K.B. Client-edge-cloud hierarchical federated learning. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9148862> (accessed on 19 August 2023).
62. Shi, W.; Zhou, S.; Niu, Z.; Jiang, M.; Geng, L. Joint device scheduling and resource allocation for latency constrained wireless federated learning. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 453–467. [CrossRef]
63. Lim, W.Y.B.; Ng, J.S.; Xiong, Z.; Jin, J.; Zhang, Y.; Niyato, D.; Leung, C.; Miao, C. Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 536–550. [CrossRef]
64. Asad, M.; Otoum, S.; Shaukat, S. Resource and Heterogeneity-aware Clients Eligibility Protocol in Federated Learning. In Proceedings of the GLOBECOM 2022-2022 IEEE Global Communications Conference, 2022; pp. 1140–1145. Available online: <https://ieeexplore.ieee.org/document/10000884/> (accessed on 19 August 2023).
65. Li, Q.; He, B.; Song, D. Model-contrastive federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021; pp. 10713–10722. Available online: https://openaccess.thecvf.com/content/CVPR2021/html/Li_Model-Contrastive_Federated_Learning_CVPR_2021_paper.html (accessed on 19 August 2023).
66. Amiri, M.M.; Gündüz, D.; Kulkarni, S.R.; Poor, H.V. Update aware device scheduling for federated learning at the wireless edge. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), 2020; pp. 2598–2603. Available online: <https://ieeexplore.ieee.org/document/9173960/> (accessed on 19 August 2023).
67. Wang, T.; Liu, Y.; Zheng, X.; Dai, H.N.; Jia, W.; Xie, M. Edge-based communication optimization for distributed federated learning. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 2015–2024. [CrossRef]
68. Li, A.; Zhang, L.; Tan, J.; Qin, Y.; Wang, J.; Li, X.Y. Sample-level data selection for federated learning. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021; pp. 1–10. Available online: <https://ieeexplore.ieee.org/document/9488723> (accessed on 19 August 2023).
69. Deng, Y.; Lyu, F.; Ren, J.; Wu, H.; Zhou, Y.; Zhang, Y.; Shen, X. Auction: Automated and quality-aware client selection framework for efficient federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 1996–2009. [CrossRef]
70. Shyu, C.R.; Putra, K.T.; Chen, H.C.; Tsai, Y.Y.; Hossain, K.T.; Jiang, W.; Shae, Z.Y. A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications. *Appl. Sci.* **2021**, *11*, 11191.
71. Hu, K.; Wu, J.; Weng, L.; Zhang, Y.; Zheng, F.; Pang, Z.; Xia, M. A novel federated learning approach based on the confidence of federated Kalman filters. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3607–3627. [CrossRef]
72. Lewy, D.; Mańdziuk, J.; Ganzha, M.; Paprzycki, M. StatMix: Data augmentation method that relies on image statistics in federated learning. In Proceedings of the Neural Information Processing: 29th International Conference, ICONIP 2022, Virtual Event, 22–26 November 2022; pp. 574–585.

73. Tang, M.; Ning, X.; Wang, Y.; Sun, J.; Wang, Y.; Li, H.; Chen, Y. FedCor: Correlation-based active client selection strategy for heterogeneous federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 10102–10111. Available online: https://openaccess.thecvf.com/content/CVPR2022/html/Tang_FedCor_Correlation-Based_Active_Client_Selection_Strategy_for_Heterogeneous_Federated_Learning_CVPR_2022_paper.html (accessed on 19 August 2023).
74. Sultana, A.; Haque, M.M.; Chen, L.; Xu, F.; Yuan, X. Eiffel: Efficient and fair scheduling in adaptive federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 4282–4294. [[CrossRef](#)]
75. Liu, S.; Chen, Q.; You, L. Fed2a: Federated learning mechanism in asynchronous and adaptive modes. *Electronics* **2022**, *11*, 1393. [[CrossRef](#)]
76. Chen, Y.; Ning, Y.; Slawski, M.; Rangwala, H. Asynchronous online federated learning for edge devices with non-iid data. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), 2020; pp. 15–24. Available online: <https://ieeexplore.ieee.org/document/9378161/> (accessed on 19 August 2023).
77. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
78. Huang, T.; Lin, W.; Wu, W.; He, L.; Li, K.; Zomaya, A.Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1552–1564. [[CrossRef](#)]
79. Shen, G.; Gao, D.; Yang, L.; Zhou, F.; Song, D.; Lou, W.; Pan, S. Variance-reduced heterogeneous federated learning via stratified client selection. *arXiv* **2022**, arXiv:2201.05762.
80. Ma, Z.; Zhao, M.; Cai, X.; Jia, Z. Fast-convergent federated learning with class-weighted aggregation. *J. Syst. Archit.* **2021**, *117*, 102125. [[CrossRef](#)]
81. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. *arXiv* **2020**, arXiv:2002.06440.
82. Haddadpour, F.; Mahdavi, M. On the convergence of local descent methods in federated learning. *arXiv* **2019**, arXiv:1910.14425.
83. Li, C.; Li, G.; Varshney, P.K. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet Things J.* **2021**, *9*, 1136–1147. [[CrossRef](#)]
84. Lee, S.; Sahu, A.K.; He, C.; Avestimehr, S. Partial model averaging in federated learning: Performance guarantees and benefits. *arXiv* **2022**, arXiv:2201.03789.
85. Beaussart, M.; Grimberg, F.; Hartley, M.A.; Jaggi, M. Waffle: Weighted averaging for personalized federated learning. *arXiv* **2021**, arXiv:2110.06978.
86. Giuseppe, A.; Manfredi, S.; Pietrabissa, A. A weighted average consensus approach for decentralized federated learning. *Mach. Intell. Res.* **2022**, *19*, 319–330. [[CrossRef](#)]
87. Chen, J.; Li, J.; Huang, R.; Yue, K.; Chen, Z.; Li, W. Federated learning for bearing fault diagnosis with dynamic weighted averaging. In Proceedings of the 2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), 2021; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9670854> (accessed on 19 August 2023).
88. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [[CrossRef](#)]
89. Kholod, I.; Yanaki, E.; Fomichev, D.; Shalugin, E.; Novikova, E.; Filippov, E.; Nordlund, M. Open-source federated learning frameworks for IoT: A comparative review and analysis. *Sensors* **2020**, *21*, 167. [[CrossRef](#)] [[PubMed](#)]
90. Poli, C. An Adaptive Model Averaging Procedure for Federated Learning (AdaFed). *J. Adv. Inf. Technol.* **2022**, *13*, 539–548.
91. Wang, S.; Suwandi, R.C.; Chang, T.H. Demystifying model averaging for communication-efficient federated matrix factorization. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021; pp. 3680–3684. Available online: <https://ieeexplore.ieee.org/document/9413927> (accessed on 19 August 2023).
92. Ji, S.; Saravirta, T.; Pan, S.; Long, G.; Walid, A. Emerging trends in federated learning: From model fusion to federated x learning. *arXiv* **2021**, arXiv:2102.12920.
93. Liang, P.P.; Liu, T.; Ziyin, L.; Allen, N.B.; Auerbach, R.P.; Brent, D.; Salakhutdinov, R.; Morency, L.P. Think locally, act globally: Federated learning with local and global representations. *arXiv* **2020**, arXiv:2001.01523.
94. Hanzely, F.; Richtárik, P. Federated learning of a mixture of global and local models. *arXiv* **2020**, arXiv:2002.05516.
95. Luping, W.; Wei, W.; Bo, L. CMFL: Mitigating communication overhead for federated learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019; pp. 954–964. Available online: <https://ieeexplore.ieee.org/document/8885054> (accessed on 19 August 2023).
96. Zhang, L.; Shen, L.; Ding, L.; Tao, D.; Duan, L.Y. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 10174–10183. Available online: https://openaccess.thecvf.com/content/CVPR2022/html/Zhang_Fine-Tuning_Global_Model_via_Data-Free_Knowledge_Distillation_for_Non-IID_Federated_CVPR_2022_paper.html (accessed on 19 August 2023).
97. Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; Guo, S. A learning-based incentive mechanism for federated learning. *IEEE Internet Things J.* **2020**, *7*, 6360–6368. [[CrossRef](#)]
98. Wink, T.; Nocht, Z. An approach for peer-to-peer federated learning. In Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2021; pp. 150–157. Available online: <https://ieeexplore.ieee.org/document/9502443/> (accessed on 19 August 2023).
99. Lalitha, A.; Kilinc, O.C.; Javidi, T.; Koushanfar, F. Peer-to-peer federated learning on graphs. *arXiv* **2019**, arXiv:1901.11173.

100. Mills, J.; Hu, J.; Min, G. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet Things J.* **2019**, *7*, 5986–5994. [CrossRef]
101. Liu, Y.; Garg, S.; Nie, J.; Zhang, Y.; Xiong, Z.; Kang, J.; Hossain, M.S. Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach. *IEEE Internet Things J.* **2020**, *8*, 6348–6358. [CrossRef]
102. Ding, J.; Tramel, E.; Sahu, A.K.; Wu, S.; Avestimehr, S.; Zhang, T. Federated learning challenges and opportunities: An outlook. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022; pp. 8752–8756. Available online: <https://ieeexplore.ieee.org/document/9746925> (accessed on 19 August 2023).
103. Haddadpour, F.; Kamani, M.M.; Mokhtari, A.; Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2021; pp. 2350–2358. Available online: <https://proceedings.mlr.press/v130/haddadpour21a.html> (accessed on 19 August 2023).
104. Zhao, Z.; Xia, J.; Fan, L.; Lei, X.; Karagiannidis, G.K.; Nallanathan, A. System optimization of federated learning networks with a constrained latency. *IEEE Trans. Veh. Technol.* **2021**, *71*, 1095–1100. [CrossRef]
105. Chen, M.; Yang, Z.; Saad, W.; Yin, C.; Poor, H.V.; Cui, S. Performance optimization of federated learning over wireless networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), 2019; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9013160> (accessed on 19 August 2023).
106. Al-Shedivat, M.; Gillenwater, J.; Xing, E.; Rostamizadeh, A. Federated learning via posterior averaging: A new perspective and practical algorithms. *arXiv* **2020**, arXiv:2010.05273.
107. Gao, H.; Thai, M.T.; Wu, J. When Decentralized Optimization Meets Federated Learning. *IEEE Netw.* **2023**, early access.
108. Wang, Z.; Xu, H.; Liu, J.; Huang, H.; Qiao, C.; Zhao, Y. Resource-efficient federated learning with hierarchical aggregation in edge computing. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021; pp. 1–10. Available online: <https://ieeexplore.ieee.org/document/9488756> (accessed on 19 August 2023).
109. Balakrishnan, R.; Akdeniz, M.; Dhakal, S.; Himayat, N. Resource management and fairness for federated learning over wireless edge networks. In Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020; pp. 1–5. Available online: <https://ieeexplore.ieee.org/document/9154285> (accessed on 19 August 2023).
110. Balasubramanian, V.; Aloqaily, M.; Reisslein, M.; Scaglione, A. Intelligent resource management at the edge for ubiquitous IoT: An SDN-based federated learning approach. *IEEE Netw.* **2021**, *35*, 114–121. [CrossRef]
111. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019; pp. 1–7. Available online: <https://ieeexplore.ieee.org/document/8761315> (accessed on 19 August 2023).
112. Trindade, S.; Bittencourt, L.F.; da Fonseca, N.L. Management of resource at the network edge for federated learning. *arXiv* **2021**, arXiv:2107.03428.
113. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A survey on federated learning for resource-constrained IoT devices. *IEEE Internet Things J.* **2021**, *9*, 1–24. [CrossRef]
114. Victor, N.; Alazab, M.; Bhattacharya, S.; Magnusson, S.; Maddikunta, P.K.R.; Ramana, K.; Gadekallu, T.R. Federated Learning for IoUT: Concepts, Applications, Challenges and Opportunities. *arXiv* **2022**, arXiv:2207.13976.
115. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated learning in edge computing: A systematic survey. *Sensors* **2022**, *22*, 450. [CrossRef]
116. Yang, H.H.; Liu, Z.; Quek, T.Q.; Poor, H.V. Scheduling policies for federated learning in wireless networks. *IEEE Trans. Commun.* **2019**, *68*, 317–333. [CrossRef]
117. Wadu, M.M.; Samarakoon, S.; Bennis, M. Joint client scheduling and resource allocation under channel uncertainty in federated learning. *IEEE Trans. Commun.* **2021**, *69*, 5962–5974. [CrossRef]
118. Hu, C.H.; Chen, Z.; Larsson, E.G. Device scheduling and update aggregation policies for asynchronous federated learning. In Proceedings of the 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2021; pp. 281–285. Available online: <https://ieeexplore.ieee.org/document/9593194> (accessed on 19 August 2023).
119. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M.; Poor, H.V.; Cui, S. Delay minimization for federated learning over wireless communication networks. *arXiv* **2020**, arXiv:2007.03462.
120. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef] [PubMed]
121. Albasyoni, A.; Safaryan, M.; Condat, L.; Richtárik, P. Optimal gradient compression for distributed and federated learning. *arXiv* **2020**, arXiv:2010.03246.
122. Ozkara, K.; Singh, N.; Data, D.; Diggavi, S. Quped: Quantized personalization via distillation with applications to federated learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3622–3634.
123. Jiang, Y.; Wang, S.; Valls, V.; Ko, B.J.; Lee, W.H.; Leung, K.K.; Tassiulas, L. Model pruning enables efficient federated learning on edge devices. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, early access.
124. Prakash, P.; Ding, J.; Chen, R.; Qin, X.; Shu, M.; Cui, Q.; Guo, Y.; Pan, M. IoT Device Friendly and Communication-Efficient Federated Learning via Joint Model Pruning and Quantization. *IEEE Internet Things J.* **2022**, *9*, 13638–13650. [CrossRef]

125. Jiang, Z.; Xu, Y.; Xu, H.; Wang, Z.; Qiao, C.; Zhao, Y. Fedmp: Federated learning through adaptive model pruning in heterogeneous edge computing. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), 2022; pp. 767–779. Available online: <https://ieeexplore.ieee.org/document/9835327> (accessed on 19 August 2023).
126. Wu, C.; Wu, F.; Lyu, L.; Huang, Y.; Xie, X. Communication-efficient federated learning via knowledge distillation. *Nat. Commun.* **2022**, *13*, 2032. [[CrossRef](#)]
127. Yuan, X.; Li, P. On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10752–10765.
128. Pappas, C.; Chatzopoulos, D.; Lalis, S.; Vavalis, M. Ipls: A framework for decentralized federated learning. In Proceedings of the 2021 IFIP Networking Conference (IFIP Networking), 2021; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9472790/> (accessed on 19 August 2023).
129. Das, A.; Patterson, S. Multi-tier federated learning for vertically partitioned data. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021; pp. 3100–3104. Available online: <https://ieeexplore.ieee.org/document/9415026> (accessed on 19 August 2023).
130. Romanini, D.; Hall, A.J.; Papadopoulos, P.; Titcombe, T.; Ismail, A.; Cebere, T.; Sandmann, R.; Roehm, R.; Hoeh, M.A. Pyvertical: A vertical federated learning framework for multi-headed splitnn. *arXiv* **2021**, arXiv:2104.00489.
131. Huang, W.; Li, T.; Wang, D.; Du, S.; Zhang, J.; Huang, T. Fairness and accuracy in horizontal federated learning. *Inf. Sci.* **2022**, *589*, 170–185. [[CrossRef](#)]
132. Su, L.; Lau, V.K. Hierarchical federated learning for hybrid data partitioning across multitype sensors. *IEEE Internet Things J.* **2021**, *8*, 10922–10939. [[CrossRef](#)]
133. Zhang, X.; Yin, W.; Hong, M.; Chen, T. Hybrid federated learning: Algorithms and implementation. *arXiv* **2020**, arXiv:2012.12420.
134. Khan, L.U.; Pandey, S.R.; Tran, N.H.; Saad, W.; Han, Z.; Nguyen, M.N.; Hong, C.S. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Commun. Mag.* **2020**, *58*, 88–93. [[CrossRef](#)]
135. Nguyen, V.D.; Sharma, S.K.; Vu, T.X.; Chatzinotas, S.; Ottersten, B. Efficient federated learning algorithm for resource allocation in wireless IoT networks. *IEEE Internet Things J.* **2020**, *8*, 3394–3409. [[CrossRef](#)]
136. Cho, Y.J.; Wang, J.; Joshi, G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv* **2020**, arXiv:2010.01243.
137. AbdulRahman, S.; Tout, H.; Mourad, A.; Talhi, C. FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IEEE Internet Things J.* **2020**, *8*, 4723–4735. [[CrossRef](#)]
138. Alferaidi, A.; Yadav, K.; Alharbi, Y.; Viriyasitavat, W.; Kautish, S.; Dhiman, G. Federated Learning Algorithms to Optimize the Client and Cost Selections. *Math. Probl. Eng.* **2022**, *2022*, 8514562. [[CrossRef](#)]
139. Imteaj, A.; Amini, M.H. FedPARL: Client activity and resource-oriented lightweight federated learning model for resource-constrained heterogeneous IoT environment. *Front. Commun. Netw.* **2021**, *2*, 657653. [[CrossRef](#)]
140. Xia, W.; Wen, W.; Wong, K.K.; Quek, T.Q.; Zhang, J.; Zhu, H. Federated-learning-based client scheduling for low-latency wireless communications. *IEEE Wirel. Commun.* **2021**, *28*, 32–38. [[CrossRef](#)]
141. Wadu, M.M.; Samarakoon, S.; Bennis, M. Federated learning under channel uncertainty: Joint client scheduling and resource allocation. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9120649/> (accessed on 19 August 2023).
142. Asad, M.; Moustafa, A.; Ito, T. FedOpt: Towards communication efficiency and privacy preservation in federated learning. *Appl. Sci.* **2020**, *10*, 2864. [[CrossRef](#)]
143. Yu, R.; Li, P. Toward resource-efficient federated learning in mobile edge computing. *IEEE Netw.* **2021**, *35*, 148–155. [[CrossRef](#)]
144. Zhou, Y.; Pu, G.; Ma, X.; Li, X.; Wu, D. Distilled one-shot federated learning. *arXiv* **2020**, arXiv:2009.07999.
145. Lin, T.; Kong, L.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 2351–2363.
146. Zhu, J.; Li, S.; You, Y. Sky Computing: Accelerating Geo-distributed Computing in Federated Learning. *arXiv* **2022**, arXiv:2202.11836.
147. Guberović, E.; Lipić, T.; Čavrak, I. Dew intelligence: Federated learning perspective. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 2021; pp. 1819–1824. Available online: <https://ieeexplore.ieee.org/document/9529852> (accessed on 19 August 2023).
148. Qu, L.; Zhou, Y.; Liang, P.P.; Xia, Y.; Wang, F.; Adeli, E.; Fei-Fei, L.; Rubin, D. Rethinking architecture design for tackling data heterogeneity in federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 10061–10071. Available online: https://openaccess.thecvf.com/content/CVPR2022/html/Qu_Rethinking_Architecture_Design_for_Tackling_Data_Heterogeneity_in_Federated_Learning_CVPR_2022_paper.html (accessed on 19 August 2023).
149. Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; Feng, J. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 5972–5984.
150. Zeng, M.; Wang, X.; Pan, W.; Zhou, P. Heterogeneous Training Intensity for Federated Learning: A Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Sci. Eng.* **2022**, *10*, 990–1002. [[CrossRef](#)]
151. Mitra, A.; Jaafar, R.; Pappas, G.J.; Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14606–14619.

152. Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; Ding, Z.; Chen, C. Local learning matters: Rethinking data heterogeneity in federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 8397–8406. Available online: https://openaccess.thecvf.com/content/CVPR2022/html/Mendieta_Local_Learning_Matters_Rethinking_Data_Heterogeneity_in_Federated_Learning_CVPR_2022_paper.html (accessed on 19 August 2023).
153. Li, Y.; Zhou, W.; Wang, H.; Mi, H.; Hospedales, T.M. Fedh2l: Federated learning with model and statistical heterogeneity. *arXiv* **2021**, arXiv:2101.11296.
154. Ma, X.; Zhu, J.; Lin, Z.; Chen, S.; Qin, Y. A state-of-the-art survey on solving non-IID data in Federated Learning. *Future Gener. Comput. Syst.* **2022**, *135*, 244–258. [[CrossRef](#)]
155. Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; Zhang, Y. Personalized cross-silo federated learning on non-iid data. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021; Volume 35, pp. 7865–7873. Available online: <https://ojs.aaai.org/index.php/AAAI/article/view/16960> (accessed on 19 August 2023).
156. Yeganeh, Y.; Farshad, A.; Navab, N.; Albarqouni, S. Inverse distance aggregation for federated learning with non-iid data. In Proceedings of the Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, 4–8 October 2020; pp. 150–159.
157. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
158. Li, Q.; Diao, Y.; Chen, Q.; He, B. Federated learning on non-iid data silos: An experimental study. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), 2022; pp. 965–978. Available online: <https://ieeexplore.ieee.org/document/9835537/> (accessed on 19 August 2023).
159. Wang, D.; Shen, L.; Luo, Y.; Hu, H.; Su, K.; Wen, Y.; Tao, D. FedABC: Targeting Fair Competition in Personalized Federated Learning. *arXiv* **2023**, arXiv:2302.07450.
160. Li, A.; Sun, J.; Wang, B.; Duan, L.; Li, S.; Chen, Y.; Li, H. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv* **2020**, arXiv:2008.03371.
161. Yu, S.; Nguyen, P.; Abebe, W.; Qian, W.; Anwar, A.; Jannesari, A. Spatl: Salient parameter aggregation and transfer learning for heterogeneous clients in federated learning. *arXiv* **2021**, arXiv:2111.14345.
162. Ruan, Y.; Zhang, X.; Liang, S.C.; Joe-Wong, C. Towards flexible device participation in federated learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2021; pp. 3403–3411. Available online: <https://proceedings.mlr.press/v130/ruan21a.html> (accessed on 19 August 2023).
163. Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; Alvarez, J.M. Personalized federated learning with first order model optimization. *arXiv* **2020**, arXiv:2012.08565.
164. Yu, L.; Albelaihi, R.; Sun, X.; Ansari, N.; Devetsikiotis, M. Jointly optimizing client selection and resource management in wireless federated learning for internet of things. *IEEE Internet Things J.* **2021**, *9*, 4385–4395. [[CrossRef](#)]
165. Cheng, Y.; Lu, J.; Niyato, D.; Lyu, B.; Kang, J.; Zhu, S. Federated transfer learning with client selection for intrusion detection in mobile edge computing. *IEEE Commun. Lett.* **2022**, *26*, 552–556. [[CrossRef](#)]
166. Pillutla, K.; Malik, K.; Mohamed, A.R.; Rabbat, M.; Sanjabi, M.; Xiao, L. Federated learning with partial model personalization. In Proceedings of the International Conference on Machine Learning, 2022; pp. 17716–17758. Available online: <https://proceedings.mlr.press/v162/pillutla22a.html> (accessed on 19 August 2023).
167. Jiang, J.; Hu, L. Decentralised federated learning with adaptive partial gradient aggregation. *CAAI Trans. Intell. Technol.* **2020**, *5*, 230–236. [[CrossRef](#)]
168. Asad, M.; Aslam, M.; Jilani, S.F.; Shaukat, S.; Tsukada, M. SHFL: K-Anonymity-Based Secure Hierarchical Federated Learning Framework for Smart Healthcare Systems. *Future Internet* **2022**, *14*, 338. [[CrossRef](#)]
169. Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Li, P.; Guo, S. A survey of incentive mechanism design for federated learning. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1035–1044. [[CrossRef](#)]
170. Zeng, R.; Zeng, C.; Wang, X.; Li, B.; Chu, X. A comprehensive survey of incentive mechanism for federated learning. *arXiv* **2021**, arXiv:2106.15406.
171. Toyoda, K.; Zhang, A.N. Mechanism design for an incentive-aware blockchain-enabled federated learning platform. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), 2019; pp. 395–403. Available online: <https://ieeexplore.ieee.org/document/9006344> (accessed on 19 August 2023).
172. Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; Zhang, J. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet Things J.* **2019**, *6*, 10700–10714. [[CrossRef](#)]
173. Han, J.; Khan, A.F.; Zawad, S.; Anwar, A.; Angel, N.B.; Zhou, Y.; Yan, F.; Butt, A.R. Tiff: Tokenized incentive for federated learning. In Proceedings of the 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), 2022; pp. 407–416. Available online: <https://ieeexplore.ieee.org/document/9860652> (accessed on 19 August 2023).
174. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Lyu, L.; Liu, Y. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1817–1829. [[CrossRef](#)]
175. Kim, S. Incentive design and differential privacy based federated learning: A mechanism design perspective. *IEEE Access* **2020**, *8*, 187317–187325. [[CrossRef](#)]

176. Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; Yang, Q. A fairness-aware incentive scheme for federated learning. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 393–399. Available online: https://dl.acm.org/doi/abs/10.1145/3375627.3375840?casa_token=I7BkjRl2ITMAAAAAA:j8480Q_PSQfMpfVnzX5U2GZhlKKfihAgPMo8uq49Vr34IA0IUTMDoRVpXHY3AA_MF2qkzu5FD3Qew (accessed on 19 August 2023).
177. Wang, X.; Zhao, Y.; Qiu, C.; Liu, Z.; Nie, J.; Leung, V.C. Infedge: A blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3325–3342. [CrossRef]
178. Jayaram, K.; Muthusamy, V.; Thomas, G.; Verma, A.; Purcell, M. Adaptive Aggregation For Federated Learning. *arXiv* **2022**, arXiv:2203.12163.
179. Tan, L.; Zhang, X.; Zhou, Y.; Che, X.; Hu, M.; Chen, X.; Wu, D. AdaFed: Optimizing Participation-Aware Federated Learning with Adaptive Aggregation Weights. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2708–2720. [CrossRef]
180. Sun, W.; Lei, S.; Wang, L.; Liu, Z.; Zhang, Y. Adaptive federated learning and digital twin for industrial internet of things. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5605–5614. [CrossRef]
181. Wang, Y.; Lin, L.; Chen, J. Communication-efficient adaptive federated learning. In Proceedings of the International Conference on Machine Learning, 2022; pp. 22802–22838. Available online: <https://proceedings.mlr.press/v162/wang22o.html> (accessed on 19 August 2023).
182. Zhou, P.; Fang, P.; Hui, P. Loss tolerant federated learning. *arXiv* **2021**, arXiv:2105.03591.
183. Andreina, S.; Marson, G.A.; Möllering, H.; Karame, G. Baffle: Backdoor detection via feedback-based federated learning. In Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021; pp. 852–863. Available online: <https://ieeexplore.ieee.org/document/9546463/> (accessed on 19 August 2023).
184. Nguyen, N.H.; Nguyen, P.L.; Nguyen, T.D.; Nguyen, T.T.; Nguyen, D.L.; Nguyen, T.H.; Pham, H.H.; Truong, T.N. FedDRL: Deep Reinforcement Learning-based Adaptive Aggregation for Non-IID Data in Federated Learning. In Proceedings of the 51st International Conference on Parallel Processing, 2022; pp. 1–11. Available online: https://dl.acm.org/doi/abs/10.1145/3545008.3545085?casa_token=ki3sb1BKfhcAAAAA:G99Gr9CAcdW3uWG4jQaQbFQICM4J4jEkmr0swtY8VFPptSVZH-oRcGY6nJXZHDpw-10_5Aggh18o_w (accessed on 19 August 2023).
185. Zhang, J.; Guo, S.; Qu, Z.; Zeng, D.; Zhan, Y.; Liu, Q.; Akerkar, R. Adaptive federated learning on non-iid data with resource constraint. *IEEE Trans. Comput.* **2021**, *71*, 1655–1667. [CrossRef]
186. Buyukates, B.; Ulukus, S. Timely communication in federated learning. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9484497/> (accessed on 19 August 2023).
187. Sharma, I.; Sharma, A.; Gupta, S.K. Asynchronous and Synchronous Federated Learning-based UAVs. In Proceedings of the 2023 Third International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2023; pp. 105–109. Available online: <https://ieeexplore.ieee.org/document/10044951> (accessed on 19 August 2023).
188. Caldas, S.; Konečný, J.; McMahan, H.B.; Talwalkar, A. Expanding the reach of federated learning by reducing client resource requirements. *arXiv* **2018**, arXiv:1812.07210.
189. Oh, Y.; Lee, N.; Jeon, Y.S.; Poor, H.V. Communication-efficient federated learning via quantized compressed sensing. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 1087–1100. [CrossRef]
190. Moustafa, A.; Asad, M.; Shaikat, S.; Norta, A. Ppca: Partial participation-based compressed and secure aggregation in federated learning. In Proceedings of the Advanced Information Networking and Applications: Proceedings of the 35th International Conference on Advanced Information Networking and Applications (AINA-2021), 2021; Volume 2, pp. 345–357. Available online: https://link.springer.com/chapter/10.1007/978-3-030-75075-6_28 (accessed on 19 August 2023).
191. Shah, S.M.; Lau, V.K. Model compression for communication efficient federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, early access.
192. Li, Y.; He, Z.; Gu, X.; Xu, H.; Ren, S. AFedAvg: Communication-efficient federated learning aggregation with adaptive communication frequency and gradient sparse. *J. Exp. Theor. Artif. Intell.* **2022**, 1–23. [CrossRef]
193. Kumar, G.; Toshniwal, D. Neuron Specific Pruning for Communication Efficient Federated Learning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022; pp. 4148–4152. Available online: https://dl.acm.org/doi/abs/10.1145/3511808.3557658?casa_token=ChA7OHSjH8wAAAAA:dBSDxTud31f78I4p9B4XmkEjqTcZf24IOL06M9I0UMFXIqUPx7VRHAYnyU-c5VmFWd_6rOim8Dlew (accessed on 19 August 2023).
194. Wu, X.; Yao, X.; Wang, C.L. FedSCR: Structure-based communication reduction for federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1565–1577. [CrossRef]
195. Qiu, X.; Fernandez-Marques, J.; Gusmao, P.P.; Gao, Y.; Parcollet, T.; Lane, N.D. ZeroFL: Efficient on-device training for federated learning with local sparsity. *arXiv* **2022**, arXiv:2208.02507.
196. Yao, D.; Pan, W.; O'Neill, M.J.; Dai, Y.; Wan, Y.; Jin, H.; Sun, L. Fedhm: Efficient federated learning for heterogeneous models via low-rank factorization. *arXiv* **2021**, arXiv:2111.14655.
197. Zhou, H.; Cheng, J.; Wang, X.; Jin, B. Low rank communication for federated learning. In Proceedings of the Database Systems for Advanced Applications. DASFAA 2020 International Workshops: BDMS, SeCoP, BDQM, GDMA, and AIDE, Jeju, Republic of Korea, 24–27 September 2020; pp. 1–16.
198. Hartebrodt, A.; Röttger, R.; Blumenthal, D.B. Federated singular value decomposition for high dimensional data. *arXiv* **2022**, arXiv:2205.12109.

199. Hu, Y.; Sun, X.; Tian, Y.; Song, L.; Tan, K.C. Communication Efficient Federated Learning with Heterogeneous Structured Client Models. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *7*, 753–767. [[CrossRef](#)]
200. Huang, J.; Tong, Z.; Feng, Z. Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization. *Int. J. Commun. Syst.* **2022**, e5161. [[CrossRef](#)]
201. Alsulaimawi, Z. A non-negative matrix factorization framework for privacy-preserving and federated learning. In Proceedings of the 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020; pp. 1–6. Available online: <https://ieeexplore.ieee.org/document/9287113> (accessed on 19 August 2023).
202. Li, M.; Andersen, D.G.; Smola, A.J.; Yu, K. Communication efficient distributed machine learning with the parameter server. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
203. Asad, M.; Moustafa, A.; Aslam, M. CEEP-FL: A comprehensive approach for communication efficiency and enhanced privacy in federated learning. *Appl. Soft Comput.* **2021**, *104*, 107235. [[CrossRef](#)]
204. Li, S.; Qi, Q.; Wang, J.; Sun, H.; Li, Y.; Yu, F.R. GGS: General gradient sparsification for federated learning in edge computing. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020; pp. 1–7. Available online: <https://ieeexplore.ieee.org/document/9148987> (accessed on 19 August 2023).
205. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, *5*, 1–19. [[CrossRef](#)]
206. Qiao, Y.; Munir, M.S.; Adhikary, A.; Raha, A.D.; Hong, S.H.; Hong, C.S. A Framework for Multi-Prototype Based Federated Learning: Towards the Edge Intelligence. In Proceedings of the 2023 International Conference on Information Networking (ICOIN), 2023; pp. 134–139. Available online: <https://ieeexplore.ieee.org/document/10048999> (accessed on 19 August 2023).
207. Asad, M.; Shaukat, S.; Javanmardi, E.; Nakazato, J.; Tsukada, M. A Comprehensive Survey on Privacy-Preserving Techniques in Federated Recommendation Systems. *Appl. Sci.* **2023**, *13*, 6201. [[CrossRef](#)]
208. Larasati, H.T.; Firdaus, M.; Kim, H. Quantum Federated Learning: Remarks and Challenges. In Proceedings of the 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2022; pp. 1–5. Available online: <https://ieeexplore.ieee.org/document/9842983> (accessed on 19 August 2023).
209. Dai, S.; Meng, F. Addressing modern and practical challenges in machine learning: A survey of online federated and transfer learning. *Appl. Intell.* **2022**, *53*, 11045–11072. [[CrossRef](#)]
210. Keçeci, C.; Shaqfeh, M.; Mbayed, H.; Serpedin, E. Multi-Task and Transfer Learning for Federated Learning Applications. *arXiv* **2022**, arXiv:2207.08147.
211. Tam, P.; Corrado, R.; Eang, C.; Kim, S. Applicability of Deep Reinforcement Learning for Efficient Federated Learning in Massive IoT Communications. *Appl. Sci.* **2023**, *13*, 3083. [[CrossRef](#)]
212. Liu, B.; Lv, N.; Guo, Y.; Li, Y. Recent Advances on Federated Learning: A Systematic Survey. *arXiv* **2023**, arXiv:2301.01299.
213. Zhou, S.; Li, G.Y. FedGiA: An efficient hybrid algorithm for federated learning. *IEEE Trans. Signal Process.* **2023**, *71*, 1493–1508. [[CrossRef](#)]
214. Yang, T.J.; Xiao, Y.; Motta, G.; Beaufays, F.; Mathews, R.; Chen, M. Online Model Compression for Federated Learning with Large Models. *arXiv* **2022**, arXiv:2205.03494.
215. Ahmed, S.T.; Kumar, V.V.; Singh, K.K.; Singh, A.; Muthukumar, V.; Gupta, D. 6G enabled federated learning for secure IoMT resource recommendation and propagation analysis. *Comput. Electr. Eng.* **2022**, *102*, 108210. [[CrossRef](#)]
216. Rajasekaran, A.S.; Maria, A.; Rajagopal, M.; Lorincz, J. Blockchain enabled anonymous privacy-preserving authentication scheme for internet of health things. *Sensors* **2022**, *23*, 240. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.