Journal on Artificial Intelligence

Tech Science Press

DOI: 10.32604/jai.2024.049912

ARTICLE

Design Pattern and Challenges of Federated Learning with Applications in Industrial Control System

Hina Batool[1], **Jiuyun Xu**[1,*], **Ateeq Ur Rehman**[2] **and Habib Hamam**[3,4,5,6]

[1]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, 266580, China

[2]School of Computing, Gachon University, Seongnam, 13120, Korea

[3]Department of Electrical and Electronic Engineering, University of Johannesburg, Johannesburg, 2006, South Africa

[4]Faculty of Engineering, Uni de Moncton, Moncton, NB E1A3E9, Canada

[5]Hodmas University College, Taleh Area, Mogadishu, Somalia

[6]International Bridges for Academic Excellence, Tunis, Tunisia

*Corresponding Author: Jiuyun Xu. Email: jyxu@upc.edu.cn

Received: 22 January 2024 Accepted: 25 March 2024 Published: 06 May 2024

ABSTRACT

Federated Learning (FL) appeared as an encouraging approach for handling decentralized data. Creating a FL system needs both machine learning (ML) knowledge and thinking about how to design system software. Researchers have focused a lot on the ML side of FL, but have not paid enough attention to designing the software architecture. So, in this survey, a set of design patterns is described to tackle the design issues. Design patterns are like reusable solutions for common problems that come up when designing software architecture. This paper focuses on (1) design patterns such as architectures, frameworks, client selection protocols, personalization techniques, and model aggregation techniques that are building blocks of the FL system. It inquires about trade-offs and working principles accompanying each design aspect, providing insights into their effect on the scalability, performance, or security process; (2) elaborates challenges faced in the design and execution of FL systems such as communication efficiency, statistical/system heterogeneity, or security/privacy concerns. It additionally investigates continuous exploration efforts and distinguishes future examination headings to take out the design challenges and upgrade the adequacy of the frameworks, and (3) depicts some FL applications used in industrial control systems along with their limitations that pave a new research gap for industry professionals. This comprehensive study provides a valuable resource for researchers, practitioners, and system designers interested in understanding the design aspects and challenges associated with FL.

KEYWORDS

Machine learning; federated learning; Internet of Things (IoT); industrial control system; FL challenges

1 Introduction

In the past years especially after artificial intelligence (AI) surpassed humans in the board game Alpha-Go [1], Machine Learning (ML), and AI have gained popularity. Adoption of ML in multiple domains like healthcare, e-commerce, finance, smart home, and customer services applications further accelerated due to the availability of large amounts of powerful computing units and Big Data. Traditionally, in many ML applications, cloud platforms or central servers collect and aggregate data from multiple devices and organizations for training the models [2]. This centralized approach has its limitations, particularly when the training dataset encompasses confidential information, posing potential security risks. Additionally, concerns regarding data ownership, confidentiality, user privacy, and the introduction of new data usage and management regulations like the general data protection regulation (GDPR) necessitate the need for fair, private, efficient, and secure distributed ML model training [3]. Federated learning (FL) is an evolving technology [4] that has garnered significant interest among researchers due to its potential and applicability. It enables algorithms to gain experience, a feature that is not always assured with conventional ML approaches [5]. Fig. 1 provides a visual representation of the process. FL typically consists of four main steps, as depicted below.

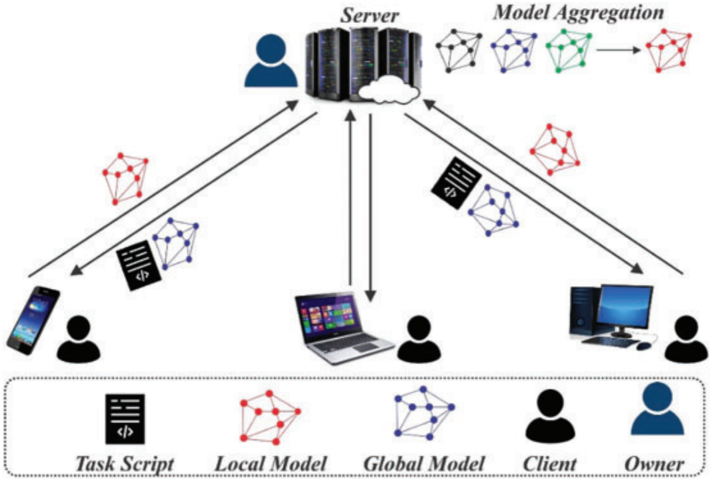

Figure 1: Simplified FL framework

1.1 Client Selection

During this stage, the server chooses participants from a group of devices, employing either random selection or specific algorithms.

1.2 Parameter Broadcasting

Subsequently, the initialized global model broadcasts by the server, denoted as w_G^0, to the selected participants, along with the designated task.

1.3 Training of Local Model

Every client autonomously undergoes the process of retraining the model using its local data. This training task is based on the global model w_G^t, where the current iteration index is represented for updating of local model parameters w_n^t every participant utilizes its device and data. To reduce loss function $L(w_n^t)$, every iteration t, and participant n aims to discover optimal parameters w_n^t, i.e.,

$$w_n^{t*} = \arg min_{w_n^t} L(w_n^t) \tag{1}$$

1.4 Model Aggregation

When clients complete their local model training, the updated model returns to the server. The global loss function $L(w_n^t)$ is minimized by the server through this process.

$$L\left(w_G^t\right) = \frac{1}{x}\sum\nolimits_{n=1}^{n} L(w_n^t) \tag{2}$$

The given steps are iteratively repeated for (n time), the desired number of iterations, based on the assistance from the clients to refine and improve the global model. For word auto-completion in Google keyboards, FL was primarily introduced to update language models by Google researchers [6]. In FL, data is collected from multiple sites, every participated node trains its data model, and this constructs a joint model. Notably, the shared model is encrypted no participant can approach the data of others. Moreover, the dataset of every connected participant, without being shared remains in their premises. Although the performance result of the joint model is approximately equal to the ideal centralized data-trained model, resultant privacy and security slightly differ in accuracy, particularly in explicit application domains. Furthermore, from the security and privacy advantage perspective, collaborative training in FL leads to superior models in contrast with the individually trained model by devices or organizations [7].

A FL system is a large-scale distributed system that has more design patterns and challenges. Selecting suitable parameters to meet various software quality standards and design restrictions is comparatively a difficult task. To utilize the current solutions in a better manner and encourage the adoption of FL at the corporate level, systematic guidance on the design pattern is needed. The study will help to explore answers to the following questions:

Q1: What are the emerging technologies, architectures, protocols, and frameworks used in FL design patterns?

Q2: What are the core design challenges of FL systems?

Q3: How does FL work in industrial control systems (ICS)?

In this paper, we depict a collection of various design patterns and the core vulnerabilities in the FL model. In our study, design aspects refer to FL architectures, FL frameworks, aggregation techniques, and client selection protocols that are involved in each step of the federated model. Moreover, core challenges: Communication efficiency, client scheduling, and selection, statistical and system heterogeneity, security, and privacy in the FL model are also discussed with emerging techniques. The main research contributions of this paper include: It helps practitioners to understand the complete design patterns of the FL lifecycle. We discussed the core challenges related to FL that serve as a systematic guide for researchers during the development and designing of the FL model in the future. It also evaluated the solid examples of every design pattern and challenge along with multiple solutions, allowing the practitioners to better understand the working and associated vulnerabilities of the FL model. Furthermore, it provides insight into FL applications used in ICS.

After the introduction, the rest of the paper is distributed as Section 2 depicts the past studies or related work. Section 3 illustrates our major contribution as it shows discussion and analysis covering all the design patterns, and design challenges. Section 4 presentes the FL applications in ICS. In Section 5, we conclude the study.

2 Related Work

This segment analyzes and investigates the most contemporary survey papers containing the design aspects and challenges related to FL. We looked at a lot of research on FL, focusing on the data side of applications. We searched for articles from 2018 to 2024 in databases like IEEE Xplore, Science Direct, and PubMed. Table 1 shows the summary of combining different methods that deal with uneven data, hoping it helps other researchers. Reference [8] examined how FL varies from typical distributed ML. Also, they talked about FL's novel challenges and characteristics, alongside its future scope and current methods. Conversely, in each turn of communication, compression techniques decrease the message size. Active sampling techniques are employed to influence or select participating devices based on associated overhead or system resources [9]. Another study [10] concentrates on ME networks, addressing crucial challenges including security/privacy concerns, system heterogeneity, and costly communication. In the context of communication cost challenges, various approaches were discussed, such as updating to prioritize local model update and selective gradients, model compression, and local updating, which specifically focuses on end and edge computing. Similarly, in a different work [11], the local sub-problem of every participant undergoes correction to constrain the effect of local updates, ensuring they remain faster than the primary global model.

Table 1: Comparative analysis of different approaches

	Design patterns					Design challenges				
Reference	FL architecture	FL frameworks	Aggregation techniques	Personalization techniques	Client selection protocol	Communication efficiency	Client scheduling and selection	System heterogeneity	Statistical heterogeneity	Security/privacy
[12]	✓	✓	✓	×	✓	✓	✓	✓	✓	×
[13]	×	✓	×	✓	×	✓	✓	✓	✓	✓
[14]	×	×	✓	✓	×	✓	×	×	✓	✓
[15]	✓	×	✓	×	✓	✓	×	✓	✓	✓
[16]	✓	✓	×	✓	×	✓	×	✓	✓	✓
[17]	✓	×	✓	×	✓	×	×	✓	✓	✓
[18]	×	✓	✓	✓	✓	✓	✓	×	×	✓
[19]	✓	×	×	×	×	✓	×	✓	×	✓
[20]	✓	✓	×	✓	×	×	×	✓	✓	✓
[21]	✓	✓	✓	✓	×	×	✓	×	×	×
This survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

In the study presented in [22], the focus is on adjusting each client's local gradient as a parameter by bringing the predictable mean closer to the median of gradients, it enhances the process of gradient aggregation. Similarly, in [23], the parameter under correction pertains to the neurons within the neural network (NN), aiming to facilitate size adaptation for the global model. Furthermore, reference [24] reduces the local model divergence by exploring the hyperparameters correction. In the subsequent sections, they delve into each of these approaches, providing a more detailed explanation for clarity. To address challenges related to privacy and security, various measures are employed. Information-exploiting attacks are mitigated through the use of techniques such as Generative Adversarial Network (GAN) model training [25], selective participant inclusion, Differential Privacy (DP), secret sharing schemes, and selective parameter sharing techniques. Researchers in [26] demonstrate that the FedAvg algorithm can achieve satisfactory accuracy even in scenarios where the data is non-ID among applicants. Specifically, it is observed [27] that a CNN-trained model for the CIFAR-10 dataset which exhibits FedAvg gives 51% less accuracy as compared to a centralized trained model. This decrease in accuracy is quantified using the Earth Mover's Distance (EMD), which measures the disparity between the data distribution of participants in FL and the overall population distribution. For the specific models, Kashin's illustration [4] is used instead of the Hadamard transformation to incorporate a structured random rotation before quantization and subsampling. Kashin's representation has proven to be more suitable in this context.

In [28], the task of FL using wireless channels is considered and a fair-wireless federated learning (FWFL) algorithm that specifically focus on imperfections in these channels. The goal was to create a model that is fair to all users and works satisfactorily for everyone. This was done by treating FL like a problem of managing multiple objectives and making some changes to the classic MGDA algorithm. Experiments showed that FWFL performs even better when there is a high level of differences among users. Also, in some situations, the average accuracy improves because it is designed to handle noise, unlike the comparison methods. Research [29] describes architectures of FL according to domain usage, such as FEDF (use when privacy has to be preserved in a parallel trained distributed environment), vertical FL, federated transfer learning (FTL), FedHealth, horizontal FL, deep learning (DL), federated autonomus deep learning (FADL), Blockchain FL, FTL architecture and PerFit for cloud-connected devices. Moreover, the research also focuses on multiple aggregation techniques such as FL-based stochastic controlled averaging (Scaffold), FedProx, FedBCD, FedCS, SMC-avg, FedAvg, federated distillation, FAug, VerifyNet, LoAdaBoost, PrivFL, and HybridFL. From different vantage points, FL's constraints and difficulties share commonalities, and indeed, there could be further challenges and constraints emerging when implementing FL in such scenarios [30]. Some shared hurdles and limitations encompass data diversity, security concerns, accountability and traceability issues, performance matters, system architecture, privacy and security considerations, and the trust factor [31]. While FL aims to safeguard privacy, it cannot address all privacy concerns completely. In [32], a new way of doing FL called FedCache is specifically designed for making personalized edge devices shrewder. FedCache sets up storage space on the central server to save new knowledge sent by individual devices. It then recovers personalized information related to precise private data by looking at similar patterns in the stored knowledge. Based on this, it improves and refines the local models on the devices to make them personalized and better. Experiments show that FedCache achieves similar accuracy compared to the best methods in personalized FL while reducing the amount of communication needed. A new technique called Split Federated Learning (SFL) offers better performance than current collaborative learning approaches. SL and FL are explained as collaborative learning methods along upcoming 6G networks and their main goals and timelines. Studies emphasize the importance of SFL for the future 6G networks, considering various aspects like 6G use cases, available datasets and frameworks, and technologies [33].

3 Discussion and Analysis

This section explains the complete design patterns of a federated environment. The design aspect consists of architectures of FL, frameworks for FL, client selection protocols, aggregation techniques, and personalization techniques as illustrated in Fig. 2. In the remaining section, we explain challenges related to FL. The Challenges section encompasses key obstacles, including communication efficiency, client selection, statistical variation, security or privacy concerns, and system diversity.

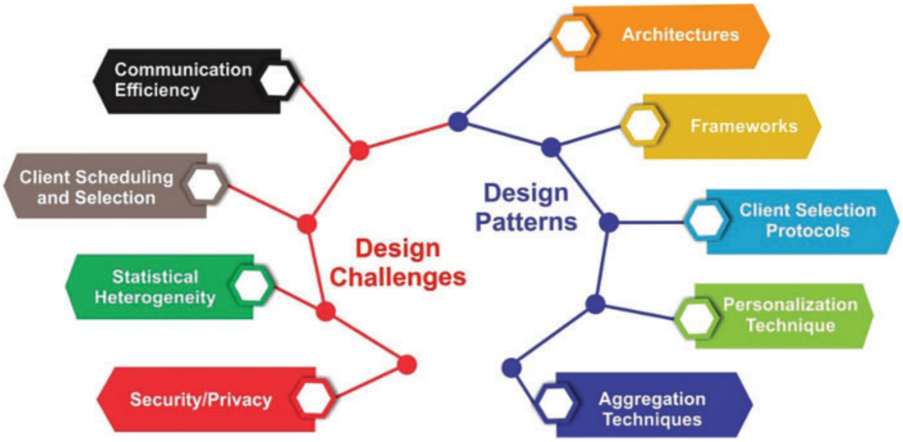

Figure 2: Framework patterns and challenges of FL

3.1 Design Patterns

The first pillar is FL architectures: We discuss HFL, VFL, FTL, FEDF, Cross-Device FL, PerFit, MMVFL, Cross-Silo FL, FADL, and E-HER for FL. In the second point, we discuss FL frameworks: PySyft, TFF, LEAF, FATE, CrypTen, FFL-ERL, and Tensor/IO. All these frameworks have their specification and challenges. Third, we briefly explain the client selection protocol FedCS and other most commonly used protocols like hybrid-FL and VerifyNet. In the last section, we explain the aggregation techniques listed as FedProx, FedAvg, FedMA, FedPAQ, TurboAgg, HireFAVG, SGD, FedBoost, and Scaffold.

3.1.1 Architectures

FL can be classified into primary categories depending on the partitioning of data among participants, specifically in terms of feature and sample spaces as demonstrated in Table 2.

Table 2: Descriptive analysis of FL architecture

Architecture	Objective	Challenges
HFL	Collaborative model training across devices with local data sharing and global model aggregation	Data heterogeneity
VFL	Collaboration between data owners with different data attributes	Data privacy; security
FTL	Utilizing pre-trained models and techniques to adapt and transfer knowledge	Domain adaptation; model transfer
FEDF	Extracting shared features from distributed data sources	Communication efficiency; latency
CFL	Training models collaboratively across devices	Device and network heterogeneity

(Continued)

Table 2 (continued)

Architecture	Objective	Challenges
PerFit	Customizing global models to individual participants' preferences	Personalization; model aggregation
MMV-FL	Combining and learning from multiple modalities	Integration and fusion of heterogeneous modalities
CSFL	Enabling FL across different organizational silos	Data access and trust between silos
FADL	Strengthening the resilience of FL in case of adversarial attacks	Robustness to adversarial attacks

Vertical FL pertains to the training of a ML model using distinct features or attributes from various organizations, all while safeguarding data privacy [11]. Every organization possesses a unique set of characteristics, and the objective is to jointly train a model using vertically segregated data without revealing the raw data itself. Vertical FL has found applications across diverse domains, including healthcare and finance, where safeguarding data privacy is of utmost importance. An alternative FL framework has been put forth [34] designed to prioritize privacy preservation and concurrent training. Their framework, known as FEDF, enables the training of a model on multiple training datasets dispersed across different geographical locations and potentially owned by different stakeholders. Horizontal FL focuses on training a ML model [22] using the same set of features across different organizations or parties. Cross-Device FL has gained attention in applications where user data is distributed across personal devices, and privacy concerns exist. Another intriguing approach to FL is introduced in a separate work [35] where the proposed framework focuses on its suitability for Internet of Things (IoT) applications. The authors introduce the PerFit framework, designed to address certain challenges inherent in both FL and IoT as presented in Fig. 3. FTL involves transferring knowledge or model weights from a centralized model to multiple decentralized devices or parties, where each party further trains the model using its data [36]. The goal is to leverage the knowledge from the centralized model while adapting it to specific local data, enabling personalized learning without sharing raw data.

FTL allows for the combination of centralized knowledge with localized fine-tuning. Cross-Silo FL has applications in various domains, such as finance, healthcare, and telecommunications, where organizations possess different subsets of data that collectively contribute to a more comprehensive model. A different research [23] introduces a novel architecture grounded on the Vertical FL system. Precisely, the authors present the Multi-clients Multi-class Vertical FL (MMVFL) framework. This framework is designed to accommodate multiple participants, with a key feature being the secure sharing of labels from the data owner to other participants. The growth of FL has spurred a wide array of research teams and industries to explore FL for both research purposes and product development [37]. The landscape of architectures and platforms for FL continues to evolve, with additional patterns detailed in reference [38]. These platforms and architectures collectively contribute to the ongoing refinement of FL.

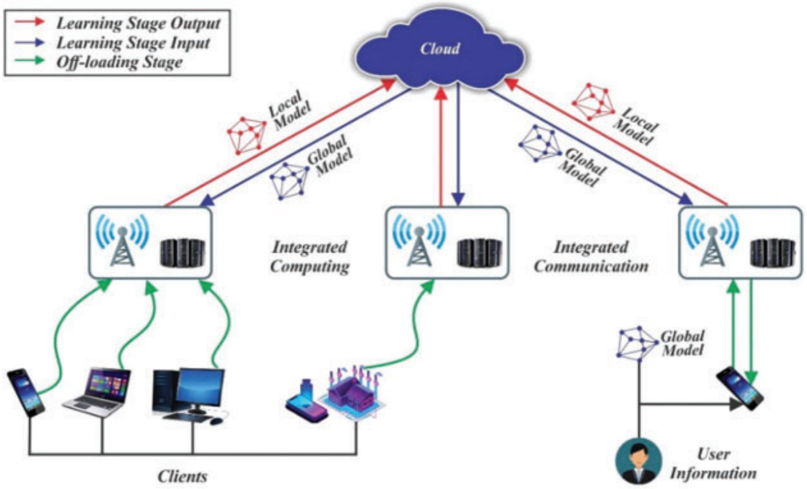

Figure 3: Overview of PerFit framework

3.1.2 Frameworks

In recent times, several open-source frameworks for FL have emerged, enlisted in Table 3. Notably, PySyft places a strong emphasis on addressing privacy issues. It achieves this by enabling the control of private data performance while training the model, all the processes occur in the Python PyTorch library [39]. The creation of SyftTensor automatically triggers a LocalTensor to do the specified command. Virtual Workers (VW) serve as the participants in a simulated FL setup. In contrast, Google's TFF [4] presents an alternative platform for FL, providing users with an open and versatile framework that can be customized to meet their precise requirements. This process is carried out in two layers: (i) FL and (ii) Federated Core. Additionally, LEAF functions as a comprehensive framework suitable for various applications, including multi-task learning, FL, and meta-learning. This platform provides access to multiple datasets for experimental purposes. The LEAF framework [40], is encompassed of three essential components: Implementation references, datasets, and metrics. WeBank's Federated AI Technology Enabler (FATE) is an open-source framework specifically crafted to support the secure and federated implementation of ML models. In contrast, CrypTen is a privacy-preserving framework developed on PyTorch [41].

Table 3: Overview of FL frameworks

Framework	Supporting software	Objectives
PySyft	Python	Maintain privacy
TFF	Tensor flow	Manage federated environment
LEAF	Python	Operate multitasking
FATE	FATE flow/board	Manage federated environment
CrypTen	PyTorch	Preserve and maintain privacy
FFL-ERL	Erlang	Control real-time parallel computing
Tensor/IO	Tensor flow	Mobile devices connectivity

CrypTen offers the additional advantage of being a library-based solution. On the other hand, FFL-ERL (Erlang) is a dynamically typed programming language with a structured framework. Introduced by [42], this platform's mathematical implementation is explained in the work. The authors conducted testing by generating an AI dataset for evaluation purposes. However, it is worth noting that FFL-ERL may incur a performance drawback. Tensor/IO [43] is a framework designed to leverage the capabilities of Tensor Flow on cell phones, including Android, React Native applications, and iOS. It offers flexibility by providing a selection of backend programming codes for consumers to choose from, such as JavaScript, Objective-C, Java, Swift, and Kotlin. The main advantage of using this framework is to make prediction easy.

3.1.3 Client Selection Protocols

In the context of FL client selection many research studies have explored multiple techniques. These studies can be characterized into five groups founded on the mechanisms employed for client selection [44]. Initially, to execute the FL task client selection procedures involved randomly choosing a small number of clients from the entire bunch of available clients. Random selection of clients can be unpredictable, so another resource-based selection approach which monitors the resources of clients was introduced to reduce dropouts. However, this might limit the number of clients. Then, incentive-based models were created to encourage clients to actively join by offering rewards in return for participation. In the quest to build effective and accurate ML models, the importance of an unbiased and reliable client selection model cannot be overstated. Fairness-based models were introduced to make sure the selection is fair and reliable, allowing a different group of clients to take part in it. Just like the FL hybrid protocol, the FedCS protocol was introduced to address challenges in FL as illustrated in Fig. 4.

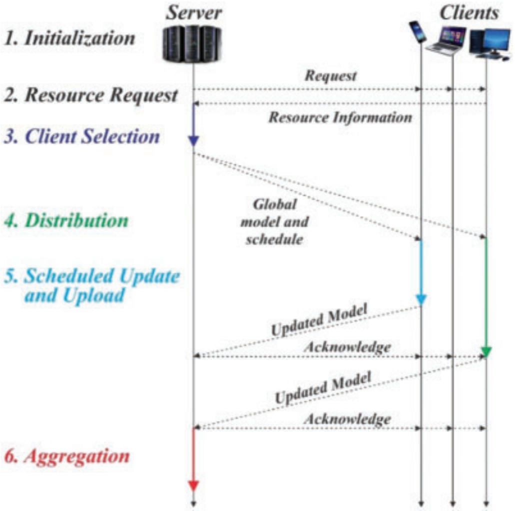

Figure 4: Overview of FedCS framework

FedMA was familiarized by [45] and is specifically tailored for advanced NN models such as Long Short-Term Memory Networks (LSTM) and Convolutional Neural Networks (CNN). In the study conducted by [46], a stochastic optimization problem addressing bandwidth allocation and joint client selection was formulated, taking into account long-term client energy constraints. The authors devised a novel algorithm called OCEAN, which utilizes real-time wireless channel information while certifying long-term performance guarantees. The results demonstrated improved accuracy, especially in non-IID dataset settings. Additionally, to alleviate the stochastic nature of client selection, a multi-criteria-based approach named FedMCCS was introduced [47].

3.1.4 Personalization Techniques

In FL, the primary objective is model training using a centralized repository without altering the samples of data. However, specific situations may necessitate personalization, allowing the customization of the global model to suit individual clients and granting users access to a more extensive model trained on a larger dataset. The process of attaining personalized predictions can be aided by integrating contextual features into datasets while maintaining privacy. To tackle the personalization challenge within FL, the research community has introduced a range of techniques. One such technique is presented by [14] who introduced a personalized FL approach. This approach provides multiple clients the facility to compress updated models through the model compression aggregation technique, thereby achieving personalized model updates while minimizing communication costs. By employing this technique, individual clients can benefit from personalized model adaptations while efficiently managing the communication overhead involved in the FL process. The comparative analysis of personalization techniques is illustrated in Table 4.

Table 4: Comparative analysis of personalization techniques

Technique	Concern	Challenges
Knowledge distillation	Controls knowledge distillation to train personalized models.	Obtaining the appropriate knowledge distillation scheme for personalized model training.
Local adaptation	Enables local adaptation of the global model by using local task-oriented data.	Ensuring efficient coordination of model updates, and managing the trade-off between global and local models.
User sampling	Creates personalized subsets of client's data for training personalized models.	To ensure representative and diverse user subsets, and address the heterogeneity in user data.
User preferences	It uses contextual bandit algorithms and adapts the model based on user feedback and preferences.	The privacy concerns related to collecting user feedback, and handling potential biases in user preferences.
Reinforcement learning	It utilizes reinforcement learning to personalize models for autonomous applications based on local data.	Dealing with limited local data on each client, addressing communication and latency issues
Transfer learning	It transfers learning to personalize the FL process on client devices.	Identification of the optimal pretraining strategy and balancing of personalized adaptation.

The system utilizes a weighted objective function to balance personalized learning objectives with privacy assurances. In the context of personalized healthcare prediction models on wearable devices, the FedHealth framework is introduced in [48] as a solution for FTL.

In [25], a strategy is designed for personalized FL, utilizing Moreau envelopes to facilitate customized model updates while assuring the protection of data privacy. FedHealth employs FTLs to ease the modification of personalized models using individual user's data. This framework validates the potential for personalized healthcare applications, underscoring the significance of wearable devices in advancing customized model training while maintaining data privacy. To validate the effectiveness of personalized FL techniques, the researchers introduced FedPerf as a standardized benchmark suite [35]. The utilization of the Moreau envelopes empowers clients to improve their models in a personalized manner while protecting the privacy of their local data.

3.1.5 Aggregation Techniques

The aggregation algorithm holds a pivotal role in composing the learning of parameters for the global model in FL. Given the privacy guarantees that prevent direct access to training pipelines and clients' data for anomaly detection, the aggregator serves as a crucial defense against potential attacks. It is accountable for incorporating suitable mechanisms to discard and identify abnormal client updates [49]. The adaptive federated optimization concept was presented by Google's research team [50] to increase the flexibility of server optimization. This approach provides flexibility by incorporating various optimization strategies on the server's side.

It is significant that FedAvg, a specific instance that uses Stochastic Gradient Descent (SGD) as both the server or client optimizer, with a server learning rate of 1, represents a specific case within adaptive federated optimization. However, it is important to acknowledge that adaptive federated optimization does not eliminate the influence of client diversity [17]. Nonetheless, it proves to be highly effective in scenarios where there is moderate and naturally occurring heterogeneity, particularly in cross-device settings. The initial framework, introduced by Google in the realm of FL is the FedAvg algorithm [1]. It is beached in the SGD optimization approach, making it principally suitable for hybrid FL models highlighting a client-server architecture.

The algorithm commences by initiating the training process, during which the server disseminates the global model parameters to a randomly selected subset of clients from a larger pool. In response to these constraints and to improve the aggregation of model updates within the framework of FL, several iterations of FedAvg have been introduced as presented in Table 5 [51]. HierFAVG [52] introduces an aggregation technique that is concerned with communication, designed to facilitate edge server level for partial model aggregation. FedBoost [53] is an FL algorithm that leverages ensemble learning techniques to achieve communication efficiency. It trains an ensemble of initially trained base predictors through FL, effectively reducing the costs associated with both client-server or server-client communications, without relying on model compression or gradient compression techniques. Scaffold [7] addresses the issue of client itinerants by retaining the technique of variance reduction in its local update. The method calculates both the update direction for the server model and each client and quantifies client divergence by examining the variance between these directions.

Table 5: Variants of FedAvg

Techniques	Concerns	Challenges
FedProx	Mitigating non-IID effects with a proximal term	Requires prior knowledge of client relationships and may lead to increased communication overhead

(Continued)

Table 5 **(continued)**

Techniques	Concerns	Challenges
FedAvg	Aggregating model updates from all clients	Sensitive to non-IID data distribution
FedMA	Active learning-based client selection to improve model training	Complexity in implementation and training
FedPAQ	Incorporating prioritized client updates based on importance	Depends on accurate importance estimation
TurboAgg	Addressing stragglers with an adaptive learning rate mechanism	Require fine-tuning for specific scenarios

3.2 Design Challenges

This section will explore five distinct design challenges associated with addressing the distributed optimization problem within the context of FL. These challenges distinguish it from other conventional problems, such as distributed learning in data center environments or traditional private data analyses. We discuss communication efficiency, client selection, statistical heterogeneity, system heterogeneity, and security/privacy. These are major vulnerabilities in FL that are based on more research directions in the future.

3.2.1 Communication Efficiency

The environment of FL consists of a significant number of devices, such as millions of cellular phones, where local computing can be more feasible than network computing, as in a network there are many issues related to resources like energy and bandwidth [54]. To tackle this issue, it is essential to consider two main factors to reduce communication overhead in the federated setting: i) diminishing the total number of communication rounds, and ii) minimizing the amount of data transmitted during each communication round as shown in Fig. 5.

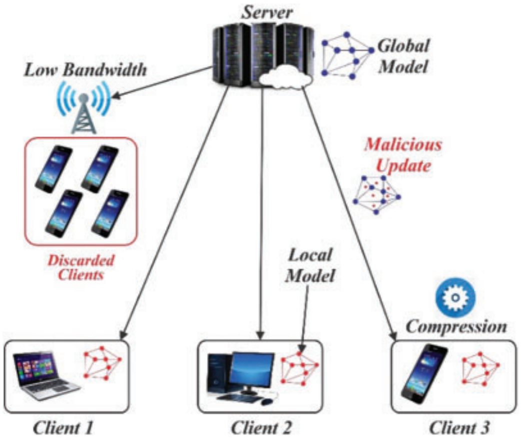

Figure 5: FL model training for various clients

Given that the model convergence in FL may require hundreds of communication rounds, the time taken for communication becomes a significant challenge. In a simplified implementation of the FL framework, each client is mandated to send a comprehensive model update to the server during each training round. The asymmetrical nature of the speed of the Internet connection implies that the downlink speed is faster than the uplink speed. To ensure compliance with the requirement that individual client updates cannot be inspected before aggregation on the server, an additional layer of security is typically implemented on top of the clients' raw updates. This, in turn, increases the data volume that needs to be uploaded [45]. Furthermore, frameworks employed to decrease communication costs, like compression, can inadvertently detrimentally impact the model's quality. Additionally, communication bottlenecks have the potential to disrupt the FL process to a significant extent as shown in Table 6.

Table 6: Literature analysis of communication efficiency

Ref.	Technique	Objective	Bottleneck
[55]	Periodic aggregation	Clients perform and synchronize numerous local updates with the server	Frequency reduction
[56]	Periodic aggregation	A collaborative pattern of communication where there is no need for a summary of local gradients	Frequency reduction
[5]	Asynchronicity	A FedAvg protocol that accommodates straggler clients	Communication type
[12]	Parameter number reduction	Clients are to upload and download only relevant parts of the full model.	Frequency reduction
[57]	Parameter number reduction	Identifies the subset of neurons in each local model	Frequency reduction
[23]	Hybrid compression	Compressing the local gradients into a finite number of bits	Size reduction

3.2.2 Client Scheduling and Selection

The core aim of this approach is to provide ease to the FL parameter server to decide which client takes part in federated tasks and how resources and training task is distributed among clients to reduce the convergence time and enhance the collaborative training performance as described in Table 7. Major client selection challenges are reliable client selection, resource management, client dropout, and client number maximization. In [58], a decentralized FL strategy is introduced, obviating the necessity for a central server. This approach employs an online Push-Sum algorithm and factors in one-way trust relationships among clients within a social network structure. The goal is to make sure it is resistant to change and nobody can deny their actions. In [3], there is an idea to use reputation and a special kind of database (blockchain) to make sure nobody can deny or mess with the FL process. In [37], they suggest adding extra information using Lagrange coding to the model updates to handle when some client dropouts occur. This extra information helps rebuild the model together if a client drops out.

Table 7: Literature of client selection techniques

Ref.	Technique	Objective	Bottleneck
[59]	Trust and reputation	Deliberates one-directional trust connections	Reliable client selection
[60]	Trust and reputation	Drive client device reputation score	Reliable client selection
[61]	Redundancy	Utilize LaGrange coding to generate a new aggregation model	Client dropout
[10]	Asynchronicity	An FL algorithm permits clients to update on time	Client dropout
[15]	Heuristics	Boosts uplink of bandwidth and allocation for power transmission	Resource management
[62]	Reinforcement learning	Deep Q-learning approach that permits the server to make ideal resource decisions	Resource management
[35]	Multi-objective optimization	Augmenting the number of participating connections and minimizing the latency	Client number maximization

3.2.3 Statistical Challenges

Statistical challenges refer to the considerations and difficulties linked to statistical inference or modeling in the privacy-preserving and distributed nature of FL as elaborated in Table 8. Some of the important statistical challenges are Non-IID Data, black cycles, Sample Size, Model heterogeneity, and bias mitigation. In [63], a specialized optimization framework for the federated environment enables personalization through the training of discrete but interconnected models for each device through multi-task learning. While this approach ensures theoretical convergence for defined objectives but is limited to convex objectives or faces scalability challenges when dealing with extensive networks. Agnostic FL [27] provides a more structured substitute by optimizing the central model by using a special technique (minimax optimization) to suit any desired distribution formed by a mix of client allocations.

In [64], a broader approach called q-FFL is proposed, which introduces an objective that assigns greater relative weight to devices with higher losses, thereby promoting reduced variance in the final accuracy distribution. Khodak et al. [52] introduce a meta-learning technique that determines the learning rate for individual devices by utilizing information from various tasks, where each task corresponds to a distinct device. Empirical results demonstrate improved performance compared to the standard FedAvg approach.

In [38], investigate a diverse solution that dynamically selects between a device-specific model and a global model to address cyclic design in data samples during federated training. Another methodology [65] is the star configuration as a Bayesian network which employs variational inference during the learning process. However, this method may encounter scalability issues when applied to large federated networks, despite its ability to handle non-convex models. In another study [66], the exploration of transfer learning for personalization involves training a global model centrally on shared proxy data and then implementing FedAvg.

Table 8: Summary of statistical challenges of FL

Ref.	Technique	Objective	Bottleneck
[67]	Data augmentation	Each client updates and trains its data according to the created dataset.	Size imbalance
[68]	Active learning	Measure the impact of local data loss, prioritize clients, signifying observations linked to minority class data.	Class imbalance
[47], [69]	Transfer learning	Considers the presence of malicious parties.	Distribution imbalance
[70]	Client clustering	Clients are clustered keeping geometric properties in mind.	Distribution imbalance
[60]	Plurality	An idea to train a variant model for every block in the FL training cycle.	Block cycles
[71]	Weighted optimization	Target distributions that consist of a blend of various client distributions.	Bias mitigation

3.2.4 System Security/Privacy

Privacy holds a prominent position in FL applications, prompting the development of several approaches to tackle this concern. FL tackles privacy issues by storing training data on individual devices, eliminating the necessity to concentrate sensitive information on a server. In contrast, centralized ML often gathers and stores data in a central location, heightening the risk to individual privacy [72] FL permits model training without sharing raw data, safeguarding privacy and supporting collaborative learning among dispersed devices. This decentralized method lessens the chances of data breaches and unauthorized access, offering a privacy-aware alternative to conventional centralized ML. Additionally, reference [47] introduces locally differentially private algorithms in the t term of meta-learning, which can be adapted for FL with personalization. These algorithms provide verifiable learning assurances in convex settings. Furthermore, the combination of differential privacy with model compression techniques proves effective in simultaneously reducing communication and preserving privacy. In [3], the focus is on detecting targeted attacks that aim to change the model's action on specific data instances. The authors introduce an apparitional anomaly detection technique that identifies and removes malicious updates and leverages the low-dimensional embedding of model updates. The underlying idea is that abnormal model. Nevertheless, the transmission of model updates throughout the training process poses a threat to sharing inner clients' details with a central server or third party. To mitigate this risk, differential privacy has been applied in FL by certain studies [73], it is offering global differential privacy guarantees. However, these approaches involve hyper-parameters that impact communication and accuracy, necessitating careful selection. In cases where heightened privacy assurances are requisite, reference [74] introduce a more lenient variant of client privacy that restricts the capabilities of strong attackers. This approach offers enhanced privacy guarantees compared to global privacy while preserving model performance.

Existing efforts focused on enhancing the privacy of FL often rely on traditional cryptographic protocols such as differential privacy and secure multiparty computation (SMC). This protocol [75] safeguard model updates in FL.

This protocol ensures that the main server is unable to directly attain the local updates but just detects the model aggregation in every round. By using SMC, the privacy guarantee is exceptionally high, preserving the original accuracy of the model. However, this method does come with a visible increment in communication overhead that can impact the overall system performance. FL is an innovative ML approach that is vulnerable to the diversity of attacks that seek to handle the collaborative learning (CL) process [76]. These security attacks can be broadly categorized into targeted attacks and untargeted attacks. Untargeted attacks, often referred to as Byzantine attacks as presented in Fig. 6, have the objective of undermining the model's performance or disrupting the overall training process, without specifically singling out any particular client or data samples, as described in Table 9. In [77], the authors examine a scenario where Byzantine clients manipulate the messages they generate before transmitting the data to the central server. They depict the RSA method, which is a type of stochastic gradient descent designed to withstand the considered Byzantine attack. In [11], a Byzantine robust FL framework named Adaptive Federated Averaging is introduced. In each iteration, HMM is employed to discard and identify updated models by accessing the analogous among aggregated model outcomes and individual updates. It enhances future iterations by blocking malicious clients from the FL system.

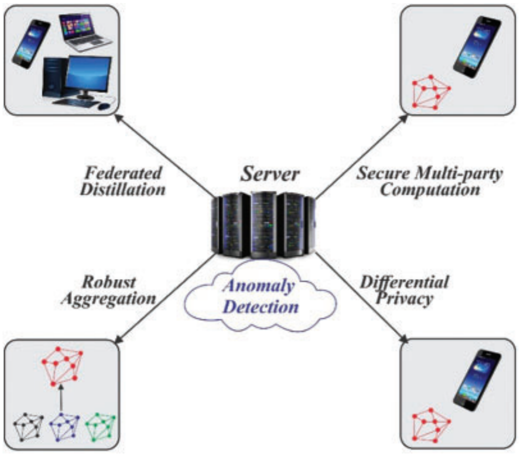

Figure 6: System security under FL

Table 9: Literature of privacy methods

Ref.	Technique	Objective	Bottleneck
[78]	Local differential privacy	Attain a tradeoff between model performance and privacy preservation.	Balancing privacy and model accuracy
[79]	Secure multiparty computation	An SMC approach that executes secure computation on client devices when the number of clients increases randomly.	Malicious attacks
[80]	Differential privacy	Ensures the privacy of individual client data.	Privacy analysis
[81]	Secure aggregation	Safeguard aggregation process against attacks.	Abnormal client updates

(Continued)

Table 9 (continued)

Ref.	Technique	Objective	Bottleneck
[82]	Homomorphic encryption	Maintain the privacy of clients.	Data confidentiality

The authors of [83] introduce Krum, a distributed ML approach that is resilient to Byzantine attacks. Krum's core concept is to ensure that the server-selected vector output aligns, on average, with the gradient direction and satisfies statistical moment bounds derived from a correct gradient estimator. In [84], the focus is on detecting targeted attacks that aim to alter the model's behavior on specific data instances. The authors introduce a spectral anomaly detection technique that leverages the low-dimensional embedding of model updates to identify and remove malicious updates. The underlying idea is that abnormal model.

4 Applications of FL in Industrial Control System

ICS as described by [85] encompass a wide range of control systems and associated instruments. These systems include devices, networks, and controls utilized for operating and automating industrial processes across various sectors such as manufacturing, transportation, energy, gas pipelines, and water treatment. ICS devices and protocols have become integral components of essential infrastructure as described in Fig. 7. Furthermore, to enhance the efficiency of industrial production processes, the industry is increasingly adopting the Industrial IoT infrastructure, which interconnects multiple intelligent physical devices. Consequently, network latency and bandwidth pose challenges when transmitting this data from distributed edge nodes to the cloud [86]. The greater the interpretability of a ML model, the easier it becomes for administrators to understand the reasons behind specific predictions. While deploying FL facilitates the efficient operation of distributed DL algorithms for anomaly detection in IoT-based ICS, it is important to note that anomaly detection techniques can only identify abnormalities.

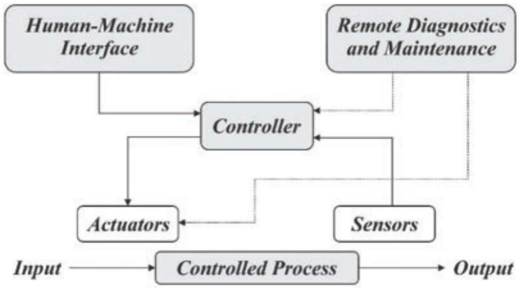

Figure 7: Flow of industrial control system

4.1 Anomaly Detection

A combined model Support Vector Data Description (SVDD) and Variational Autoencoder (VAE) with FL techniques has been developed to operate efficiently on low-powered edge devices within an IoT-based Smart Factory system [87]. This integration allows experts to swiftly analyze and respond to anomalies in distributed control system environments, offering a significant advantage in managing and troubleshooting issues effectively. Following are the limitations of anomaly detection for ICS such as there is a risk that attackers might divert transmitted information on the edge-cloud transmission medium, rather than targeting the source information medium before the edge. Additionally, the connection among manufacturing sectors, such as variations in the number of machines in each sector, presents a challenge for local learning models at the edge due to imbalances in distributed learning.

4.2 Time-Series Data Handling

FATRAF has demonstrated exceptional detection performance for time-series data in ICS when compared to state-of-the-art anomaly detection solutions. It not only delivers superior accuracy but also significantly improves processing speed, reducing the training time of the learning model to just 1200 s. This reduction opens the door for more frequent re-training of the anomaly detection solution during factory operations [88]. In the realm of ICS, the system's normal or abnormal behavior can shift over time due to factors such as device aging and changes in system configurations. This advantage underscores the effectiveness of FL in the context of FATRAF. These are the drawbacks of time-series data handling in ICS. Like devices may have diverse operating systems, processing capabilities, and data formats. Statistical variations in the data distribution across these devices make it challenging to create a uniform and effective learning model. Slow or restricted communication channels can impede the timely exchange of model updates and data.

4.3 Cyber-Attacks Detection

An edge-computing-based FL architecture tailored for intelligent applications in Smart Manufacturing within the realm of Big Data. The comprehensive solution has demonstrated superior anomaly detection performance and offers rapid response times by executing anomaly detection near the sources of potential attacks, namely, at the edge [89]. The shortcomings of cyber attack detection in ICS are: It is crucial to study the limitations concerning data size and the number of features to maintain stable operation in edge computing environments. In situations where manufacturing sites differ, such as varying machine numbers, imbalanced distributed training becomes a challenge for VAE-LSTM models running on the edge.

4.4 Security Attack Detection

GRUs are a method for precise classification and detection of attacks in IoT frameworks using FL-based anomaly detection. The approach integrates FL, leveraging on-device training to distribute computational power [90] by adding multiple layers of GRUs which enhance the precision of attack classification. Additionally, to enhance the performance of the system an ensemble is employed to combine predictions from these GRU layers. Ensuring the reliability of IoT devices [91] the privacy advantage of FL enlarges the security of the IoT framework. The evaluation shows that the current method exceeds non-FL versions of intrusion detection procedures. An evaluation environment was acknowledged, consisting of IoT devices which was tested using real-time data obtained from specific device datasets. This configuration is capable of accurately classifying liabilities linked to IoT devices, whether they are identified or unknown.

5 Conclusion

FL is chosen over traditional ML when data privacy is crucial, as it allows model training without sharing raw data. It is well-suited for decentralized, edge computing environments, enabling collaborative learning across diverse datasets on local devices. Traditional ML, often centralized, may not be practical in scenarios requiring privacy preservation and collaborative training. The FL technique involves creating a collaborative learning model where clients do some learning on their own and take part in global model training. However, this collaborative participation of mobile nodes raises some challenges, like data heterogeneity and system heterogeneity. This paper looked at the main ideas in FL and pointed out the challenges when it comes to device and data heterogeneity. It provides a comprehensive overview of the foundation of FL, encircling its protocols, frameworks, enabling technologies, and recent research addressing various aspects. This information serves as a solid foundation for data scientists seeking insight into FL technologies and protocols, offering an understanding of the distinct components and protocols that form the basis of FL. We looked closely at several studies to point out the challenges, data publicity, the limits, and suggestions. We noticed some gaps and showed all this information in tables and pictures. These details are important to help readers understand how to deal with and solve data problems in FL. Furthermore, the paper presents some of the benefits, challenges, and issues related to the design and deployment of FL. The examined applications in this study showcase how they empower industrial applications to adeptly manage extensive datasets, streamline predictive maintenance, and elevate anomaly detection capabilities. The conditions in which FL would offer the most advantages are diverse, and realizing these added benefits will necessitate ongoing development and research efforts before achieving streamlined implementation. While the frameworks and protocols discussed aim to mitigate some of FL's drawbacks, addressing systems heterogeneity remains a crucial challenge that requires further attention. Moreover, there are some software and hardware restrictions on the devices used in the FL process. Another major issue is data handling and acquisition in terms of fairness can be discussed in the future. SFL is a new technique that offers a secure and quicker way to train models. Research on SFL is still in its early stages, so it is important to explore new research paths. In the future, it can enhance the dependability of upcoming 6G systems based on SFL. Therefore, it encourages researchers to create, test, and implement creative solutions using SFL for 6G technologies.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and design: Hina Batool, Jiuyun Xu; data collection: Hina Batool, Ateeq Ur Rehman; analysis and interpretation of results: Hina Batool, Habib Hamam; draft manuscript preparation: Hina Batool; revision and suggestions: Jiuyun Xu, Ateeq Ur Rehman, Habib Hamam. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This is a review article, and no code or data was used.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017. doi: 10.1038/nature24270.

[2] Y. Song, H. H. Chang, Z. Zhou, S. Jere, and L. Liu, "Federated dynamic spectrum access," Jun. 2021. Accessed: Dec. 28, 2023. [Online]. Available: https://arxiv.org/abs/2106.14976v1

[3] K. M. Jawadur Rahman *et al.*, "Challenges, applications and design aspects of federated learning: A survey," *IEEE Access*, vol. 9, pp. 124682–124700, 2021. doi: 10.1109/ACCESS.2021.3111118.

[4] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020. doi: 10.1109/ACCESS.2020.3013541.

[5] L. Lavaur, M. O. Pahl, Y. Busnel, and F. Autrel, "The evolution of federated learning-based intrusion detection and mitigation: A survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 2309–2332, Sep. 2022. doi: 10.1109/TNSM.2022.3177512.

[6] M. Wellens, J. Riihijärvi, and P. Mähönen, "Modelling primary system activity in dynamic spectrum access networks by aggregated ON/OFF-processes," in *2009 6th IEEE Annu. Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks Work. SECON Work*, 2009. doi: 10.1109/SAHCNW.2009.5172946.

[7] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *Proc. 2019 IEEE Int. Conf. Big Data, Big Data 2019*, Dec. 2019, pp. 2597–2604. doi: 10.1109/BIGDATA47090.2019.9006179.

[8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, Aug. 2019. doi: 10.1109/MSP.2020.2975749.

[9] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020. doi: 10.1109/JIOT.2019.2956615.

[10] W. Y. Bryan Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020. doi: 10.1109/COMST.2020.2986024.

[11] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022. doi: 10.1109/JIOT.2021.3077803.

[12] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consum. Electron. Mag.*, vol. 9, no. 3, pp. 8–16, May 1, 2020. doi: 10.1109/MCE.2019.2959108.

[13] T. D. Cao, T. Truong-Huu, H. Tran, and K. Tran, "A federated deep learning framework for privacy preservation and communication efficiency," *J. Syst. Archit.*, vol. 124, no. 11, pp. 102413, Mar. 2022. doi: 10.1016/j.sysarc.2022.102413.

[14] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, no. 1, pp. 35–44, Feb. 2020. doi: 10.1109/OJCS.2020.2993259.

[15] S. Feng and H. Yu, "Multi-Participant multi-class vertical federated learning," Jan. 2020. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/2001.11154v1

[16] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "FDC: A secure federated deep learning mechanism for data collaborations in the internet of things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6348–6359, Jul. 2020. doi: 10.1109/JIOT.2020.2966778.

[17] T. Ryffel *et al.*, "A generic framework for privacy preserving deep learning," Nov. 2018. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/1811.04017v2

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, Las Vegas, NV, USA, Dec. 2015, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[19] S. Caldas *et al.*, "LEAF: A benchmark for federated settings," Dec. 2018. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/1812.01097v3

[20] J. Ren, X. Shen, Z. Lin, R. Mech, and D. J. Foran, "Personalized image aesthetics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Dec. 2017, pp. 638–647. doi: 10.1109/ICCV.2017.76.

[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," *IEEE Int. Conf. Commun.*, Shanghai, China, May 2019, pp. 1–7. doi: 10.1109/ICC.2019.8761315.

[22] Y. Liu *et al.*, "FedBCD: A communication-efficient collaborative learning framework for distributed features," *IEEE Trans. Signal Process.*, vol. 70, no. 1, pp. 4277–4290, 2022. doi: 10.1109/TSP.2022.3198176.

[23] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul. 2020. doi: 10.1109/MNET.001.1900506.

[24] H. Kim, J. Park, M. Bennis, and S. L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Aug. 12, 2018. doi: 10.1109/LCOMM.2019.2921755.

[25] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mob. Comput.*, vol. 21, no. 8, pp. 2818–2832, Dec. 2020. doi: 10.1109/TMC.2020.3045266.

[26] X. Yan, B. Cui, Y. Xu, P. Shi, and Z. Wang, "A method of information protection for collaborative deep learning under GAN model attack," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 18, no. 3, pp. 871–881, May 2021. doi: 10.1109/TCBB.2019.2940583.

[27] H. Lee, O. Simpson, M. Seol, and T. Kim, "Performance enhancement in federated learning by reducing class imbalance of Non-IID Data," *Sensors*, vol. 23, no. 3, pp. 1152, Jan. 2023. doi: 10.3390/S23031152.

[28] S. M. Hamidi and O. Damen, "Fair wireless federated learning through the identification of a common descent direction," *IEEE Commun. Lett.*, vol. 28, no. 3, pp. 567–571, 2024. doi: 10.1109/LCOMM.2024.3350378.

[29] Y. Amar, H. Haddadi, and R. Mortier, "Privacy-aware infrastructure for managing personal data," in *SIGCOMM '16: ACM SIGCOMM 2016 Conf.*, Florianopolis, Brazil, Aug. 2016, pp. 571–572. doi: 10.1145/2934872.2959054.

[30] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 911–926, 2020. doi: 10.1109/TIFS.2019.2929409.

[31] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020. doi: 10.1109/TNNLS.2019.2953131.

[32] Z. Wu *et al.*, "FedCache: A knowledge cache-driven federated learning architecture for personalized edge intelligence," Aug. 2023. Accessed: Feb. 20, 2024. [Online]. Available: https://arxiv.org/abs/2308.07816v3

[33] H. Hafi, B. Brik, P. A. Frangoudis, and A. Ksentini, "Split federated learning for 6G enabled-networks: Requirements, challenges and future directions," *IEEE Access*, vol. 12, pp. 9890–9930, Sep. 2023. doi: 10.1109/ACCESS.2024.3351600.

[34] M. Hao, H. Li, X. Luo, G. Xu, H. Yang and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Inf.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020. doi: 10.1109/TII.2019.2945367.

[35] A. Vettoruzzo, M. R. Bouguelia, and T. Rögnvaldsson, "Personalized federated learning with contextual modulation and meta-learning," Dec. 2023. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/2312.15191v1

[36] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," Feb. 2020. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/2002.05516v3

[37] J. Han *et al.*, "Heterogeneity-aware adaptive federated learning scheduling," in *Proc. 2022 IEEE Int. Conf. Big Data (Big Data)*, 2022, pp. 911–920. doi: 10.1109/BIGDATA55660.2022.10020721.

[38] S. J. Reddi *et al.*, "Adaptive federated optimization," in *ICLR 2021-9th Int. Conf. Learn. Rep.*, Feb. 2020. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/2003.00295v5.

[39] M. Orabi, J. Khalife, A. A. Abdallah, Z. M. Kassas, and S. S. Saab, "A machine learning approach for GPS code phase estimation in multipath environments," in *2020 IEEE/ION Position, Locat. Navig. Symp. (PLANS)*, Apr. 2020, pp. 1224–1229. doi: 10.1109/PLANS46316.2020.9110155.

[40] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *8th Int. Conf. Learn. Rep. ICLR 2020*, Feb. 2020. Accessed: Dec. 29, 2023. [Online]. Available: https://arxiv.org/abs/2002.06440v1.

[41] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020, pp. 1–6.

[42] A. Reisizadeh, A. Jadbabaie, A. Mokhtari, H. Hassani, and R. Pedarsani, “FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization,” in *Proc. Mach. Learn. Res.*, vol. 108, pp. 2021–2031, 2019. doi: 10.48550/arXiv.1909.13014.

[43] M. Kamp *et al.*, “Efficient decentralized deep learning by dynamic model averaging,” in *Mach. Learn. Knowl. Discov. Dat.: European Conf.*, Dublin, Ireland, 2018, pp. 393–409. doi: 10.1007/978-3-030-10925-7_24.

[44] W. Wu, L. He, W. Lin, R. Mao, C. Maple and S. Jarvis, “SAFA: A semi-asynchronous protocol for fast federated learning with low overhead,” *IEEE Trans. Comput.*, vol. 70, no. 5, pp. 655–668, May 2021. doi: 10.1109/TC.2020.2994391.

[45] H. Y. Zhu and Y. C. Jin, “Multi-objective evolutionary federated learning,” *IEEE T. Neur. Net. Lear.*, vol. 31, no. 4, pp. 1310–1322. doi: 10.1109/TNNLS.2019.2919699.

[46] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, “Asynchronous online federated learning for edge devices with non-IID data,” in *Proc. 2020 IEEE Int. Conf. Big Data, Big Data 2020*, Dec. 2020, pp. 15–24. doi: 10.1109/BIGDATA50022.2020.9378161.

[47] Y. Sun, S. Zhou, and D. Gündüz, “Energy-aware analog aggregation for federated learning with redundant data,” in *ICC 2020-2020 IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020, pp. 1–7. doi: 10.1109/ICC40277.2020.9148853.

[48] P. M. Mammen, “Federated learning: Opportunities and challenges,” 2021. doi: 10.48550/arXiv.2101.05428.

[49] S. Jing and C. Xiao, “Federated learning via over-the-air computation with statistical channel state information,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 11, pp. 9351–9365, Nov. 2022. doi: 10.1109/TWC.2022.3175887.

[50] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L. C. Wang, “Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach,” *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1345–1348, Oct. 2019. doi: 10.1109/LWC.2019.2917133.

[51] J. Wang and G. Joshi, “Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms,” *J. Mach. Learn. Res.*, vol. 22, no. 213, pp. 1–50, 2021. Accessed: Dec. 29, 2023. [Online]. Available: http://jmlr.org/papers/v22/20-147.html

[52] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019. doi: 10.1109/JIOT.2019.2940820.

[53] C. He, C. Tan, H. Tang, S. Qiu, and J. Liu, “Central server free federated learning over single-sided trust social networks,” Oct. 2019. Accessed: Dec. 29, 2023. [Online]. Available: http://arxiv.org/abs/1910.04956

[54] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi, “FedMCCS: Multicriteria client selection model for optimal IoT federated learning,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4723–4735, Mar. 2021. doi: 10.1109/JIOT.2020.3028742.

[55] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh, “Semisupervised deep reinforcement learning in support of IoT and smart city services,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018. doi: 10.1109/JIOT.2017.2712560.

[56] L. Cui *et al.*, “Joint optimization of energy consumption and latency in mobile edge computing for internet of things,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4791–4803, Jun. 2019. doi: 10.1109/JIOT.2018.2869226.

[57] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in *36th Int. Conf. Mach. Learn*, Long Beach, USA, Feb. 2019, pp. 8114–8124. Accessed: Dec. 30, 2023. [Online]. Available: https://arxiv.org/abs/1902.00146v1.

[58] S. Sharma, C. Xing, Y. Liu, and Y. Kang, “Secure and efficient federated transfer learning,” in *Proc. 2019 IEEE Int. Conf. Big Data, Big Data 2019*, Dec. 2019, pp. 2569–2576. doi: 10.1109/BIGDATA47090.2019.9006280.

[59] T. Li, M. Sanjabi, and V. Smith, Fair resource allocation in federated learning,” 2019. doi: 10.48550/arXiv.1905.10497.

[60] M. Khodak, M. Balcan, and A. Talwalkar, “Adaptive gradient-based meta-learning methods,” in *Neural Information Processing Systems*, 2019.

[61] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” 2016. doi: 10.48550/arXiv.1610.05492.

[62] C. Briggs, Z. Fan, and P. András, “Federated learning with hierarchical clustering of local updates to improve training on non-IID data,” in *2020 Int. Jt. Conf. Neural Networks (IJCNN)*, 2020, pp. 1–9. doi: 10.1109/IJCNN48605.2020.9207469.

[63] W. Liu, L. Chen, Y. Chen, and W. Zhang, “Accelerating federated learning via momentum gradient descent,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020. doi: 10.1109/TPDS.2020.2975189.

[64] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, “Collaborative deep learning in fixed topology networks,” 2017. doi: 10.48550/arXiv.1706.07880.

[65] X. Liu *et al.*, “Distributed intelligence in wireless networks,” *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1001–1039, 2023. doi: 10.1109/OJCOMS.2023.3265425.

[66] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” *IEEE Access*, vol. 8, pp. 108952–108971, 2020. doi: 10.1109/ACCESS.2020.2998358.

[67] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen and M. Chen, “In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning,” *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep. 2019. doi: 10.1109/MNET.2019.1800286.

[68] S. Feng, D. T. Niyato, P. Wang, D. I. Kim, and Y. Liang, “Joint service pricing and cooperative relay communication for federated learning,” in *2019 Int. Conf. Internet of Things (iThings) and IEEE Green Comput. Commun. (GreenCom) and IEEE Cyber, Phys. Soc. Comput. (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 815–820.

[69] E. T. M. Beltran *et al.*, “Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges,” *IEEE Commun. Surv. Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023. doi: 10.1109/COMST.2023.3315746.

[70] L. Che, J. Wang, Y. Zhou, and F. Ma, “Multimodal federated learning: A survey,” *Sensors*, vol. 23, no. 15, pp. 6986, Aug. 2023. doi: 10.3390/S23156986.

[71] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, “Digital-twin-enabled 6G: Vision, architectural trends, and future directions,” *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 74–80, Jan. 2022. doi: 10.1109/MCOM.001.21143.

[72] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020. doi: 10.1109/MNET.001.1900287.

[73] M. Du, K. Wang, Y. Chen, X. Wang, and Y. Sun, “Big data privacy preserving in multi-access edge computing for heterogeneous internet of things,” *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 62–67, Aug. 2018. doi: 10.1109/MCOM.2018.1701148.

[74] T. Nguyen and M. T. Thai, “Preserving privacy and security in federated learning,” *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 833–843, 2023. doi: 10.1109/TNET.2023.3302016.

[75] R. Al-Huthaifi, T. Li, W. Huang, J. Gu, and C. Li, “Federated learning in smart cities: Privacy and security survey,” *Inf Sci.*, vol. 632, pp. 833–857, Jun. 2023. doi: 10.1016/j.ins.2023.03.033.

[76] M. Khan, F. G. Glavin, and M. Nickles, “Federated learning as a privacy solution–An overview,” *Procedia Comput. Sci.*, vol. 217, pp. 316–325, Jan. 2023. doi: 10.1016/J.PROCS.2022.12.227.

[77] B. S. P. Thummisetti, B. S. P. Thummisetti, and H. Atluri, “Advancing healthcare informatics for empowering privacy and security through federated learning paradigms,” *Int. J. Sustain. Dev. Comput. Sci.*, vol. 1, no. 1, pp. 1–16, Jan. 2024. Accessed: Feb. 20, 2024. [Online]. Available: https://ijsdcs.com/index.php/ijsdcs/article/view/434.

[78] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor and C. S. Hong, “A crowdsourcing framework for on-device federated learning,” *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020. doi: 10.1109/TWC.2020.2971981.

[79] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," *Futur Gener. Comput. Syst.*, vol. 150, no. 6245, pp. 272–293, Jan. 2024. doi: 10.1016/j.future.2023.09.008.

[80] X. Wu, F. Huang, Z. Hu, and H. Huang, "Faster adaptive federated learning," *Artif. Intell.*, vol. 37, no. 9, pp. 10379–10387, Jun. 2023. doi: 10.1609/AAAI.V37I9.26235.

[81] X. Qiu *et al.*, "A first look into the carbon footprint of federated learning," *J. Mach. Learn. Res.*, vol. 24, no. 129, pp. 1–23, 2023. Accessed: Dec. 30, 2023. [Online]. Available: http://jmlr.org/papers/v24/21-0445.html.

[82] Y. Chen, W. Lu, X. Qin, J. Wang, and X. Xie, "MetaFed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare," *IEEE Trans. Neural Networks Learn. Syst*. doi: 10.1109/TNNLS.2023.3297103.

[83] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020. doi: 10.1109/JIOT.2020.2967772.

[84] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Inf.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020. doi: 10.1109/TII.2019.2942190.

[85] M. A. Bakar Siddique, A. Asad, R. M. Asif, A. U. Rehman, M. T. Sadiq and I. Ullah, "Implementation of incremental conductance MPPT algorithm with integral regulator by using boost converter in grid-connected PV array," *IETE J. Res.*, vol. 69, no. 6, pp. 3822–3835, 2023. doi: 10.1080/03772063.2021.1920481.

[86] S. Siddique *et al.*, "Challenges and opportunities of computational intelligence in industrial control system (ICS)," in *2023 IEEE Symp. Series Comput. Intell. (SSCI)*, 2023, pp. 1158–1163.

[87] H. T. Truong *et al.*, "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Comput. Ind.*, vol. 140, pp. 103692, Sep. 2022. doi: 10.1016/j.compind.2022.103692.

[88] M. A. B. Siddique, M. A. Khan, A. Asad, A. U. Rehman, R. M. Asif and S. U. Rehman, "Maximum power point tracking with modified incremental conductance technique in grid-connected PV array," in *CITISIA 2020-IEEE Conf. Innov. Technol. Intell. Syst. Ind. Appl. Proc.*, Nov. 2020. doi: 10.1109/CITISIA50690.2020.9371803.

[89] T. T. Huong *et al.*, "Federated learning-based explainable anomaly detection for industrial control systems," *IEEE Access*, vol. 10, pp. 53854–53872, 2022. doi: 10.1109/ACCESS.2022.3173288.

[90] Y. Xiao, H. Shao, J. Lin, Z. Huo, and B. Liu, "BCE-FL: A secure and privacy-preserving federated learning system for device fault diagnosis under non-IID condition in IIoT," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14241–14252, Apr. 15, 2024. doi: 10.1109/JIOT.2023.3340745.

[91] B. Xu, H. Zhao, H. Cao, S. Garg, G. Kaddoum and M. M. Hassan, "Edge aggregation placement for semi-decentralized federated learning in industrial internet of things," *Futur Gener. Comput. Syst.*, vol. 150, no. 2, pp. 160–170, Jan. 2024. doi: 10.1016/j.future.2023.07.035.