A systematic review of federated learning: Challenges, aggregation methods, and development tools

Badra Souhila Guendouzi, Samir Ouchani, Hiba El Assaad, Madeleine El Zaher

A Systematic Review of Federated Learning: Challenges, Aggregation Methods, and Development Tools

Badra Souhila GUENDOUZI[a], Samir OUCHANI[b], Hiba EL ASSAAD[c], Madeleine EL ZAHER[c]

[a]*LINEACT CESI, Lyon, France*
[b]*LINEACT CESI, Aix-En-Provence, France*
[c]*LINEACT CESI, Toulouse, France*

Abstract

Since its inception in 2016, federated learning has evolved into a highly promising decentralized machine learning approach, facilitating collaborative model training across numerous devices while ensuring data privacy. This survey paper offers an exhaustive and systematic review of federated learning, emphasizing its categories, challenges, aggregation techniques, and associated development tools. To commence, we outline our research strategy used for this survey and evaluate other existing reviews related to federated learning. We initiate the discussion, about federated learning concepts, with a detailed examination of the primary challenges inherent in federated learning, including communication overhead, device and data heterogeneity, and data privacy issues. Subsequently, we scrutinize and classify various aggregation techniques designed to mitigate these challenges, such as federated averaging, secure aggregation, and strategies leveraging clustering and optimization methodologies. Furthermore, we delve into the exploration of cutting-edge development tools and frameworks that expedite efficient implementations of federated learning. Through our review, we aspire to provide a holistic understanding of the federated learning landscape, thereby setting the stage for future investigations, advancements, and practical implementations in this prosperous field.

Keywords:
Federated Learning, Deep Learning, Aggregation, Privacy, Heterogeneity, IoT.

1. Introduction

The Internet of Things (IoT) has burgeoned into a paramount technology of the 21st century over recent years, profoundly impacting various domains [1]. The IoT paradigm involves the integration of intelligent sensors, actuators, rapid communication protocols, and robust cybersecurity measures to augment its functions and applications. This concept has been extrapolated to numerous other sectors, such as the Internet of Medical

Things and the Internet of Industrial Things. To manage the abundant data generated by smart devices within cyber-physical systems, IoT frameworks necessitate smart and safe big data analysis methodologies. The complexity of decisions is ever-increasing, owing to the escalating volume of data [2]. In this context, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) techniques have exhibited remarkable efficacy due to their advanced learning and processing capabilities, particularly within network-based systems. Historically, the prevalent approach to utilizing these techniques was centralized learning, in which, a single server would train the AI model using data from all connected endpoints. However, this method posed significant challenges, such as bandwidth congestion, extended data processing time, and potential privacy breaches due to the substantial data transmission from endpoints to the server [3]. Recently, this method has been superseded by distributed learning, specifically Federated Learning (FL). This approach encompasses multiple endpoints, each retaining its secure and non-distributable data [4]. These endpoints collaborate, achieving parity in learning not by sharing data but through the exchange of learning parameters.

The FL has gained significant traction in the industry because it offers several advantages over traditional ML approaches. First, it allows organizations to train ML models on data that cannot be shared due to privacy concerns, such as healthcare data [5], financial data [6], etc. Second, it enables organizations to leverage data from multiple sources to improve the accuracy of their models. Third, it reduces the cost and complexity associated with storing and managing large volumes of data. It has been adopted by several industry giants, including Google, Apple, and Microsoft. Google, for example, used the FL to improve the accuracy of its keyboard app, Gboard [4], by training the model on users' devices. Similarly, Apple used the FL to improve the accuracy of its QuickType [7] keyboard, and Microsoft used it to train a model that predicts the likelihood of a Windows machine being infected with malware.

The efficacy of FL critically depends on the proficient *aggregation* of local model updates, culminating in a global model that incorporates knowledge from all participating nodes. This key procedure not only shapes the system's performance but also its operational efficiency [4]. However, the process is compounded by complexities arising from the manifold sources of these parameters, which originate from vastly diverse local environments. These environments are characterized by a gamut of computational and communication resources, which encompass an expansive range of data collections. As such, it becomes vital to assess the adaptability of a FL framework to these variations, commonly referred to as challenges, before its deployment. These challenges can be neatly classified into four overarching categories:

1. *Costly communication*: This challenge pertains to the rate of parameter exchanges within the FL framework and its subsequent effect on system efficiency and expense.
2. *System heterogeneity*: This issue mirrors the distinct resources available on each participant's devices, thereby shaping the system's overall performance and fairness.

3. *Statistical heterogeneity*: This category encapsulates possible disparities in data distributions and learning patterns amongst different nodes.
4. *Privacy concerns*: This challenge relates to ensuring the secure transit of parameters between collaborating entities.

In this paper, we present an exhaustive review of existing aggregation methods in the FL, emphasizing their capacity to address inherent FL challenges. Our discourse commences with an overview of the FL paradigm, where we explore its approaches and challenges. Moreover, we conduct an in-depth analysis of some salient aggregation methods in FL, specifically designed to tackle these challenges and compare these methods based on their efficacy in overcoming the stated issues. Finally, we catalog tools such as benchmark datasets and frameworks for developing FL algorithms, as well as the evaluation metrics used for performance measurement. The main contributions of this review include:

1. Formulating a robust and detailed search strategy to ensure the breadth and depth of our literature review, thereby enriching the overall review process.
2. Conducting a comparative study of existing reviews and surveys on federated learning, to identify common trends, unique insights, and potential gaps in the existing body of knowledge.
3. Deliver a comprehensive introduction to federated learning, encompassing an overview, primary methodologies, and prominent challenges, thus laying a solid foundation for understanding FL.
4. Performing a detailed examination and analysis of existing FL aggregation algorithms, highlighting their strengths, weaknesses, and applications.
5. Compiling a list of significant development tools and evaluation metrics prevalent in FL, facilitating a better understanding of the practical aspects of implementing and evaluating FL models.
6. Undertaking a critical comparison of the reviewed contributions, and engaging in a thorough discussion on leading research directions in FL aggregation methods, particularly focusing on how they address its prevalent challenges.

To achieve these objectives, this paper is structured as delineated in Figure 1, with the organization unfolding as follows: **Section** 2 describes the procedures of this review. In **Section** 3, we present a selection of existing surveys on *Federated Learning*. **Section 4** provides an overview of FL and a comprehensive examination of its methodologies and categorizations. **Section 5** delves into the challenges associated with FL, and proposes solutions, incorporating references to pertinent research. An in-depth exploration of FL aggregation methods is undertaken in **Section 6**, accompanied by a comparative analysis. **Section 7** details notable datasets and frameworks that aid in the development of FL algorithms, and introduces relevant evaluation metrics for the assessment of these algorithms. The paper concludes with a discussion and envisages future directions in **Section 8**. To assist the reader, Appendix A provides a list of abbreviations utilized throughout this paper.

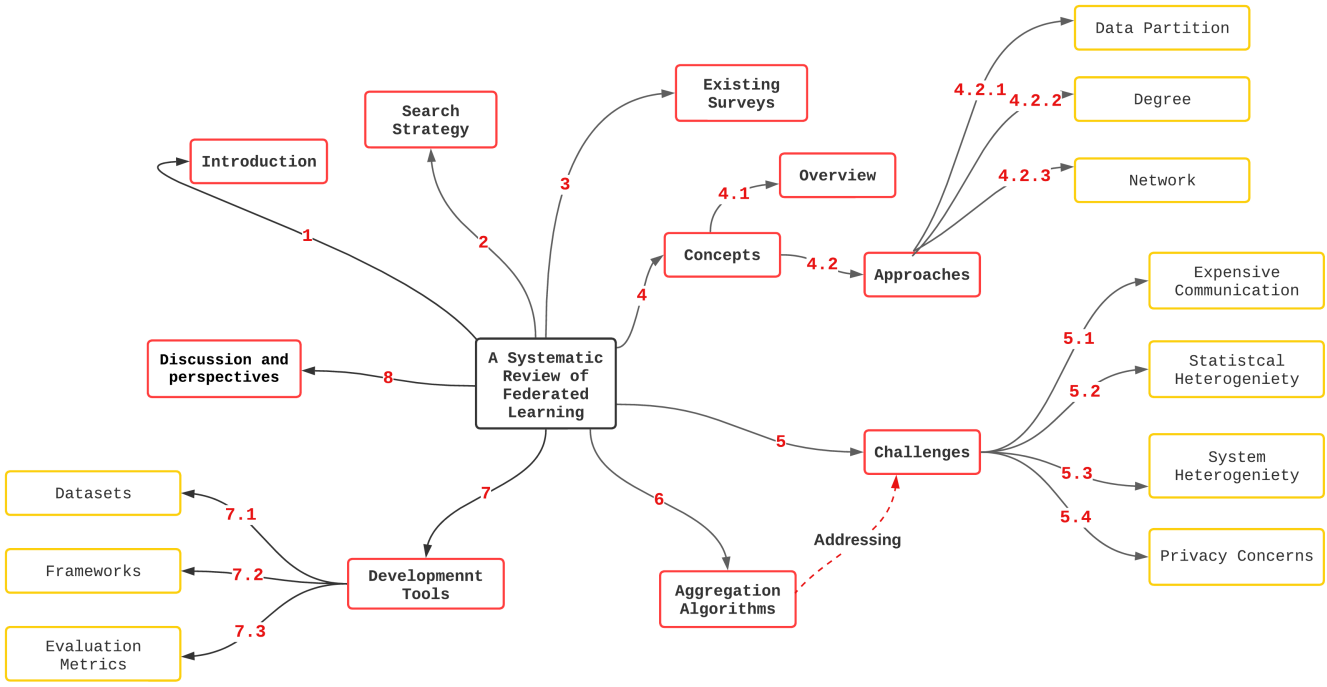

Figure 1: Paper Structure and Objectives.

2. Search Strategy

This study adopts a Systematic Literature Review (SLR) methodology [8] to amass, categorize, and scrutinize articles in the realm of the FL. In order to ascertain the inclusion of pertinent contributions within this systematic review, we executed a rigorous and expansive search, adhering to a precise sequence of steps. The formulation of our search strategy was intended to cover an extensive array of sources and databases, thus allowing for a thorough investigation of the existing literature. The exhaustive steps followed are delineated in Figure 2.

2.1. Formulation of Research Questions

As our systematic review looks to explore the realm of FL and parameter aggregation, several crucial research questions (RQs) organically surface, steering the investigation towards a holistic grasp of FL. These RQs serve as a beacon, illuminating the review process and addressing the key facets of interest. Expanding upon the challenges mentioned earlier, the following RQs have been brought to light:

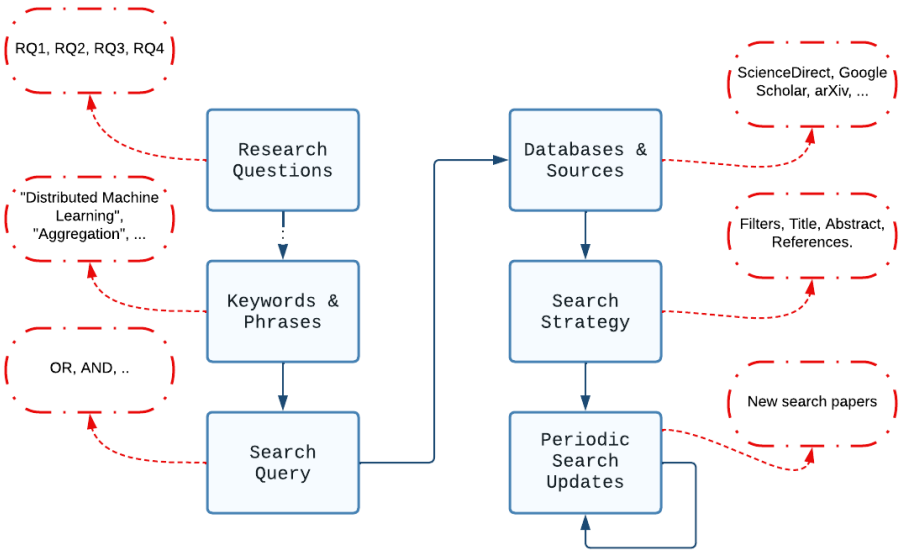

Figure 2: Our Search Strategy.

- **RQ1:** What are the key factors to be considered in designing an efficient FL framework, particularly with respect to data structure and node configuration?
- **RQ2:** How can potential challenges be effectively forecasted and mitigated throughout the FL process?
- **RQ3:** Which aggregation methodologies are optimally equipped to handle these anticipated challenges in a FL context?
- **RQ4:** What tools and techniques are best suited for validating and deploying a developed FL framework?

2.2. Identification of Keywords and Phrases

In developing a systematic review search strategy, we carefully selected keywords related to the research question, focusing on aspects of FL and its aggregation techniques. Key phrases included "Federated Learning," "Distributed Machine Learning," "Privacy-Preserving Machine Learning," "Collaborative Learning," and terms describing potential challenges. To explore data aggregation strategies in distributed learning, we considered phrases like "Aggregation Methods," "Aggregation Algorithms," and "Aggregation Techniques." For practical insights into FL implementation, we included terms such as "Development Tools," "Software," "Platforms," "FL Frameworks", and "Datasets".

2.3. Construction of the Search Query

The keywords and phrases, were synthesized using Boolean operators and parentheses to construct a comprehensive search query tailored to the context of FL and its aggregation techniques: `("Federated Learning" OR "FL" OR "Decentralized machine learning" OR "Collaborative learning") AND ("Model aggregation" OR "Parameter aggregation" OR "Multi-party computation" OR "Secure aggregation") AND ("Challenges" OR "Issues" OR "Problems" OR "Impediments") AND ("Methods" OR "Techniques" OR "Strategies") AND ("Implementation" OR "Deployment" OR "Testing tools" OR "FL frameworks").` This refined query ensures a more focused retrieval of relevant studies pertaining to the unique challenges, strategies, and tools associated with the design and implementation of FL systems and the selection of the appropriate aggregation method.

2.4. Selection of Databases and Sources

An exhaustive search was undertaken utilizing a judiciously chosen ensemble of databases and sources. These sources encompassed notable platforms such as Springer Link[1], Wiley[2], IEEE[3], and Science Direct[4], all renowned for their broad-spectrum coverage of computational and ML literature. This selection of databases was made to ensure a comprehensive search across a multitude of application areas, especially those pertaining to distributed ML, data privacy, and decentralized networks. By leveraging these diverse sources, the search strategy aspired to seize a comprehensive and diverse collection of studies germane to the domains of FL, with a focus on aggregation techniques.

2.5. Application of the Search Strategy

Our search query, tailored to the specific syntax and features of the chosen databases, was meticulously applied to ensure compatibility with FL processes. We utilized advanced search mechanisms, considering factors such as publication date, language, and other pertinent FL-related filters to refine the results. A detailed record of the search progression, including the number of relevant results from each database, was meticulously maintained to ensure transparency and reproducibility. Also, we screened the titles and abstracts of retrieved articles against the inclusion and exclusion criteria, and full-text articles meeting the criteria were further evaluated. Additionally, to guarantee comprehensive coverage, we reviewed the reference lists of relevant articles and performed citation tracking, identifying any potentially missed studies from the initial search.

[1] https://link.springer.com/
[2] https://onlinelibrary.wiley.com/
[3] https://www.ieee.org/
[4] https://www.sciencedirect.com/

2.6. Periodic Search Updates

Just before completing the systematic review, we conducted a final search to integrate the latest research findings related to the topic.

The analysis in Figure 7 indicates strong growth in FL research, emphasizing its increasing significance and wide applicability. This trend not only signifies a shift towards decentralized ML models, driven by their advantages such as data privacy and efficient resource utilization, but also places FL at the forefront of ML advancements. Particularly, the engagement with FL across diverse research fields within IEEE's conferences and journals underlines its broad implications across numerous sectors and disciplines.

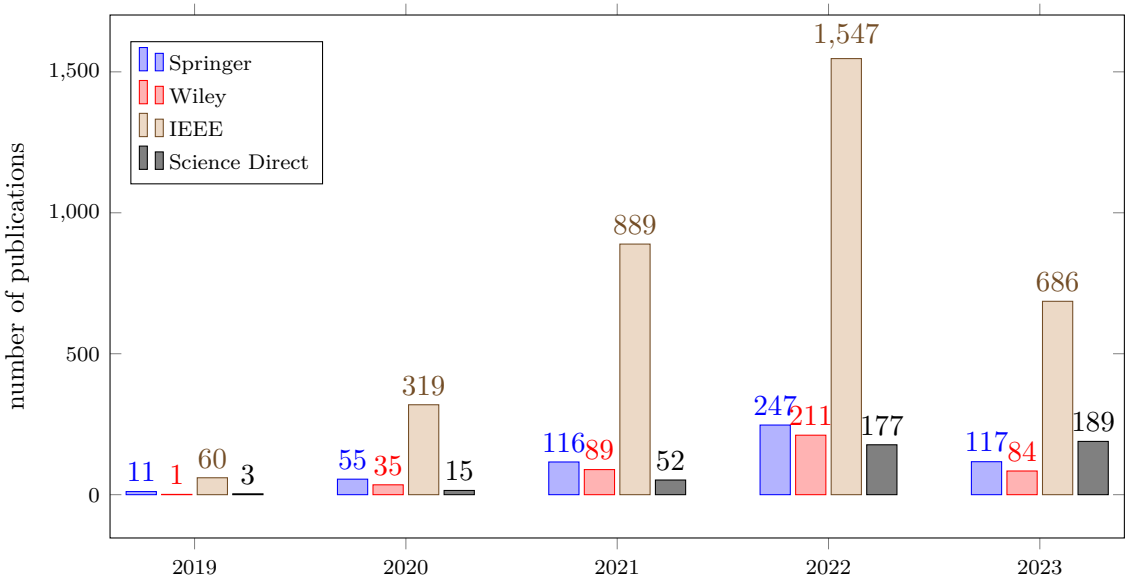

Figure 3: The number of articles related to "*Federated Learning*" in different databases visited on June 2023.

3. A Review of Existing Surveys

The domain of FL has been the subject of extensive study in a multitude of recent survey papers, each with its unique focus. According to the database **dblp** [5], we identified 84 survey papers published between 2019 and 2023 that focus on FL research. This section provides a comprehensive summary of existing surveys and reviews regarding the FL

[5] https://dblp.org/

and associated research. We have meticulously undertaken an exhaustive review of various contemporary studies, as outlined in Table 1. These investigations were scrutinized utilizing ten distinct criteria: *Background, Data, Degree, Network, Challenges, Aggregation, Surveyed Papers, Comparison, Datasets, Frameworks, Metrics*, and *Perspectives*.

1. *Background* assesses whether the survey provides the necessary foundational knowledge to comprehend FL.
2. *Data* evaluates whether the survey elucidates the data partition category within FL and cites related work.
3. *Degree* assesses whether the survey outlines the extent of FL participation.
4. *Network* considers whether the survey presents existing FL architectural models.
5. *Challenges* determines whether the survey thoroughly discusses FL's challenges.
6. *Aggregation* ascertains whether the survey referenced the various existing aggregation algorithms.
7. *Surveyed Papers* quantifies the number of papers examined in the chosen survey.
8. *Comparison* assesses whether the survey offers a comparative analysis of the reviewed papers.
9. *Datasets* verifies whether the survey lists benchmark datasets used in FL development.
10. *Frameworks* verifies whether the survey lists frameworks employed in FL development.
11. *Metrics* verifies whether the survey enumerates evaluation metrics for FL.
12. *Perspectives* evaluates whether the survey offers insights into prospective future trends and viewpoints in the field of FL.

Wen et al. 2022 [13] delivered a comprehensive overview of current FL research from five perspectives: fundamental FL knowledge, privacy and security protection mechanisms in FL, communication overhead issues, FL heterogeneity issues, and some FL applications across various fields. On the other hand, Xia et al. 2021 [9] concentrated on FL application areas (such as healthcare systems, intelligent recommendation, and vehicular network), development tools (including APIs and system designs), communication efficiency, as well as security and privacy, and migration and scheduling strategies between edge nodes and servers.

The study by Zhu et al. 2021 [10] delved into the types of data partition in FL, placing emphasis on the neural network architectures deployed in conjunction with FL. In another study, Shaheen et al. 2022 [11] explored the range of FL applications across diverse industries and domains. They examined distributed datasets and benchmarks of neural network architectures used in FL experiments and discussed the challenges posed by FL, such as statistical and system heterogeneity, data imbalance, resource allocation, and privacy concerns. On a related note, Beltrán et al. 2022 [12] investigated decentralized FL, focusing on aspects such as federation architectures, topologies, communication mechanisms, security approaches, and key performance indicators. They also analyzed and compared the

ID	Reference	Year	Publisher	Contribution
SRV1	Xia et al. [9]	2021	Elsevier	A survey of federated learning for edge computing: Research problems and solutions
SRV2	Zhu et al. [10]	2021	Springer	From federated learning to federated neural architecture search: a survey
SRV3	Shaheen et al. [11]	2022	MDPI	Applications of Federated Learning; Taxonomy, Challenges, and Research Trends
SRV4	Beltrán et al. [12]	2022	IEEE	Decentralized Federated Learning: Fundamentals, State-of-the-art, Frameworks, Trends, and Challenges
SRV5	Wen et al. [13]	2022	Springer	A survey on federated learning: challenges and applications
SRV6	Pandya et al. [14]	2023	Elsevier	Federated learning for smart cities: A comprehensive survey

Table 1: An overview of existing survey papers related to the FL research.

existing solutions to these challenges. The integration of FL with smart cities was the focal point of Pandya et al. 2023, [14], where they presented related works like the application of FL in smart transportation systems, healthcare, grid, governance, disaster management, and industries.

A comparison of these studies, as shown in Table 2, reveals that many studies tend to provide only brief or insufficient surveys of their selected criteria for reviewing FL. This finding underscores the necessity for more comprehensive and in-depth analyses of the various aspects, challenges, and aggregation algorithms pertinent to FL. Undertaking such analyses would significantly enhance our understanding and development of this field. In the given comparison, three distinct symbols are used to represent the extent to which the surveys address the specified criteria:

○ : This symbol indicates that the survey does not consider the specified criterion in its analysis.

● : Conversely, this symbol signifies that the survey fully considers and covers the specified criterion in its analysis.

◐ : This symbol represents a partial consideration of the specified criterion, meaning that the authors of the cited work have only addressed a portion of the criterion in

their analysis.

Using these symbols, we can effectively compare the scope and depth of the various survey papers, shedding light on the areas that may require further investigation or more comprehensive coverage.

ID	Background	Data	Degree	Network	Challenges	Aggregation	Surveyed papers	Comparison	Datasets	Frameworks	Metrics	Perspectives
SRV1	●	◐	●	◐	◐	◐	○	○	○	◐	○	●
SRV2	◐	●	○	○	○	◐	○	○	○	○	○	○
SRV3	●	●	○	○	◐	◐	◐	○	◐	○	○	◐
SRV4	●	◐	◐	●	◐	◐	◐	◐	◐	○	○	●
SRV5	●	●	○	○	◐	◐	◐	◐	○	○	○	●
SRV6	◐	○	○	○	◐	◐	○	○	○	○	○	●
Our	●	●	●	●	●	●	●	●	●	●	●	●

Table 2: Summary of reviews and surveys about FL.

4. Federated Learning Concepts

In this section, we aim to deliver a comprehensive overview of FL and its associated workflow process. Subsequently, to construct an effective FL framework, it becomes essential to delineate the types of each category and outline specific hyperparameters. This process enables us to identify and implement the most suitable architecture for our FL model.

4.1. Federated Learning Overview

FL is a decentralized, collaborative ML methodology, first pioneered by Google in 2016. It exemplifies a broader paradigm shift towards taking computation to the data, rather than vice versa [15]. FL effectively addresses key challenges in data privacy, ownership, and locality in scenarios where data is heterogeneous, and procured from an array of distributed devices contributing to learning. In stark contrast to conventional approaches where data is gathered first, then used to train a model, FL ensures that individual data remains with its owner and is not shared directly [16]. As described by Bonawitz et al. [17],

FL presents an ML environment in which multiple entities work in conjunction to solve an ML problem, coordinated by a central server or service provider. The raw data remains local to each client and is not transferred or exchanged. Instead, focused updates, aimed for swift aggregation, are used to realize the learning goal.

Assuming there are N devices, each trains its own local AI model with its distinct dataset D_i. Hence, FL aims to fine-tune the weight parameters $\boldsymbol{w}$ of the global model such that the loss function values for all local AI models are minimized:

$$L(w) = \frac{\sum_{i=1}^{N} |D_i| f_i(w)}{\sum_{i=1}^{N} |D_i|} \tag{1}$$

Here, f_i represents the loss function of the model trained by device $\boldsymbol{i}$ using its local dataset D_i. The fundamental structure of FL is depicted in Figure 4.

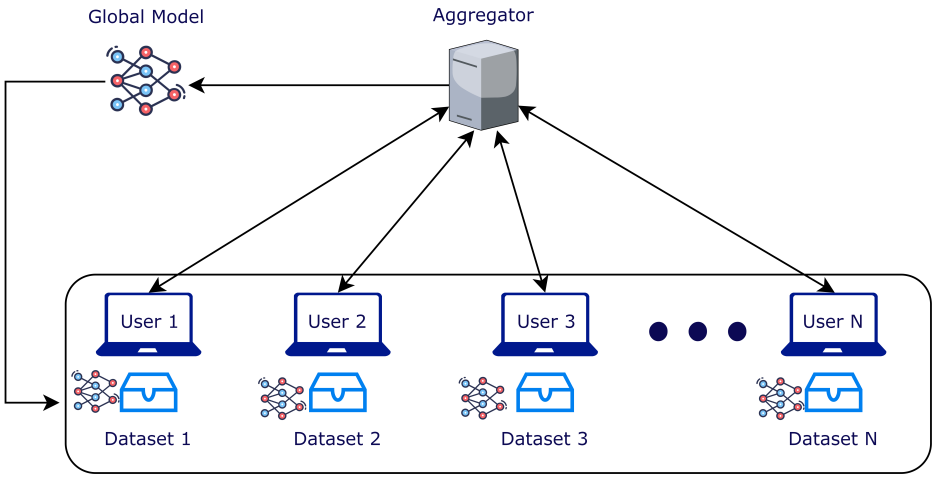

Figure 4: Basic Framework of Federated Learning.

The FL process is fundamentally aimed at the creation of an impeccable model for a specific application. The standard process flow can be outlined as follows:

1. **Problem Identification:** The initial step involves identifying the problem that we intend to solve using FL. Subsequently, we have to decide on the most fitting AI model to implement.
2. **Participant Selection:** The selection of participants relies on available online devices. These participants are chosen either randomly or through some other methodology to engage in the learning process.

3. **Training:** Every participating device begins by initializing the parameters of its ML model. It then trains this model on its private data until convergence or until a pre-determined FL communication round (CR) number is reached.
4. **Parameter Sharing:** Following local-level training, all connected participant devices *securely* transmit their model parameters to the central server using a communication method elaborated upon in Section 5.3.
5. **Parameter Aggregation:** Once the central server, which acts as the aggregator, has collected all the local models, it combines their parameters to refine and update the global model. This crucial step is further discussed in Section 6.
6. **Parameter Broadcast:** The aggregator subsequently distributes the parameters of the updated global model. The participant devices then update the parameters of their local models based on this latest information.

This process repeats until either the entire training process converges, or until a predetermined number of FL CRs have been executed (depending on the FL setup). The process is illustrated in Figure 5.

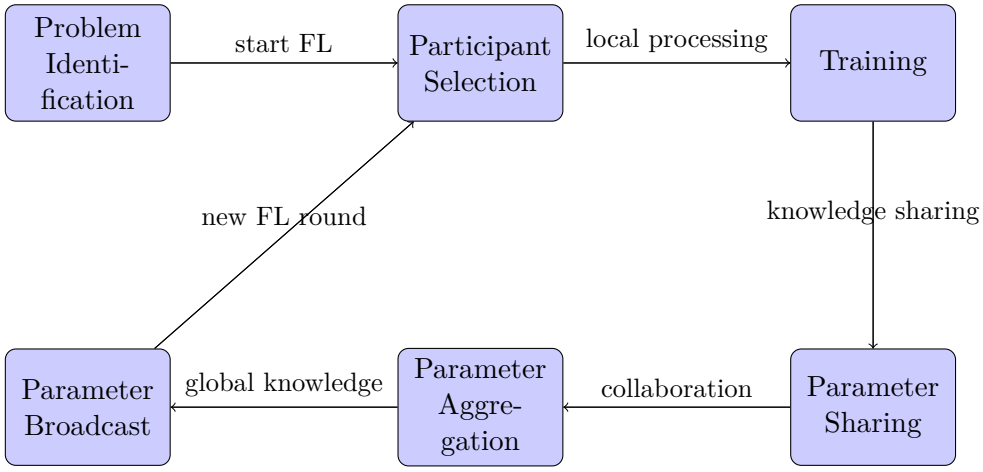

Figure 5: Illustration of the Federate Learning Workflow.

4.2. Federated Learning Approaches

FL is adopted across a plethora of domains, each with its unique combination of features, attributes, and data characteristics. This diversity contributes to the emergence of a wide array of FL architectures and types, as delineated by Du et al. [18] and depicted in Figure 6.

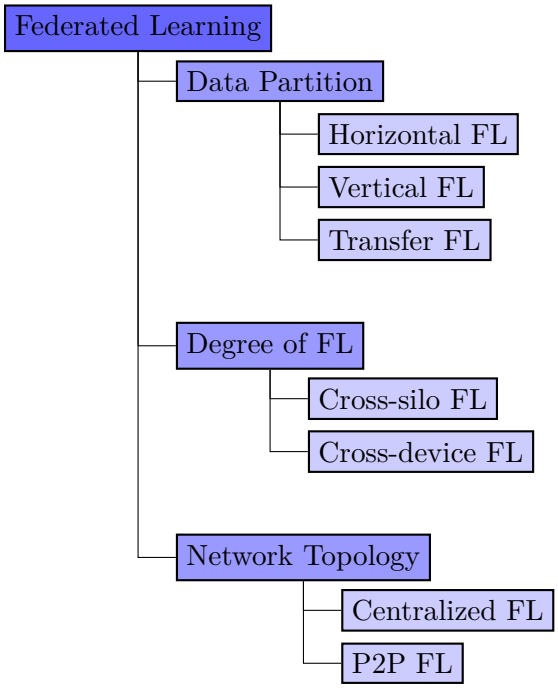

Figure 6: Federated Learning Approaches.

4.2.1. Data Partition

In this section, we delve into the classification of FL in accordance with the features of data distribution. As proposed by Yang et al. [19], disparities might exist between the feature space and the sample space of data providers. Consequently, they segmented FL into three distinct categories: Horizontally FL (HFL), Vertically FL (VFL), and Federated Transfer Learning (FTL). These categorizations are predicated on the distribution modality of data across different parties in both the feature and sample ID space. Since the inception of FL, a plethora of research endeavors have been undertaken to explore these categorizations, as depicted in Figure 7, based on data from **dblp**. In their work, Yang et al. [19] symbolize the feature space as $\mathcal{X}$, the label space as $\mathcal{Y}$, and utilize $\mathcal{I}$ to represent the sample ID space. Together, these elements constitute the comprehensive training dataset $(\mathcal{X}, \mathcal{Y}, \mathcal{I})$.

- **Horizontally Federated Learning**

  The concept of HFL, or sample-based FL, is fundamentally associated with homogeneous feature spaces. However, in situations with heterogeneous feature spaces, HFL has the limitation of only utilizing common features, thereby leaving participant-specific features unexplored. Specifically, HFL is defined for datasets that share the identical feature space $\mathcal{X}$, but are distinct in the sample ID space $\mathcal{I}$. This distinction

is mathematically represented by Equation 2.

$$\mathcal{X}_i = \mathcal{X}_j, \quad \mathcal{Y}_i = \mathcal{Y}_j, \quad \mathcal{I}_i \neq \mathcal{I}_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{2}$$

For instance, consider the scenario of data from different hospitals. Though they might share the same feature space (e.g., patient information), they diverge in their sample spaces, namely, data from different patients. As practical examples of HFL applications, Gao et al.[20] employed HFL in the training of a classification ML model using electroencephalography data derived from varied devices. Moreover, Wei et al.[21] introduced a deep flow inspection model, rooted in HFL, aimed at mitigating data congestion caused by a rapid upsurge in traffic volumes, a phenomenon which threatens the stability of the 6G network.

- **Vertically Federated Learning**

  The application of VFL, otherwise known as feature-based FL, proves suitable when dealing with scenarios where two datasets share the same sample identity space, denoted as $\mathcal{I}$, but differ concerning their feature spaces, $\mathcal{X}$. This distinction is demonstrated in Equation 3.

$$\mathcal{X}_i \neq \mathcal{X}_j, \quad \mathcal{Y}_i \neq \mathcal{Y}_j, \quad \mathcal{I}_i = \mathcal{I}_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{3}$$

  VFL demonstrates its capability to construct a robust meta-ML model by assimilating sub-models derived from a diverse set of entities. These sub-models receive their local training from data that has been partitioned vertically and exhibits varying features [22]. For instance, Zheng et al. [6] designed an FL-LRBC framework. This approach amalgamates numerous agencies to collaboratively train an optimal scorecard regression model, which is employed for credit business across a vast financial holdings group in China. Similarly, Efe [23] suggested a VFL-based multi-institutional credit scoring system aimed at enhancing the performance of ML models for industrial corporations serving a shared customer base.

- **Federated Transfer Learning**

  FTL is employed when two datasets vary within the sample ID space $\mathcal{I}$ and feature space $\mathcal{X}$, a relationship is depicted in Equation 4. Consider, for instance, two industrial firms constructing a ML model to identify industrial objects, relying on heterogeneous cameras that yield images of varying dimensions and color profiles. In such a scenario, TL could be employed to address the entirety of the sample and feature space in a federative manner. By harnessing a subset of similar samples, a common representation that bridges the two feature spaces is acquired and subsequently utilized to make predictions based on data from one source.

$$\mathcal{X}_i \neq \mathcal{X}_j, \quad \mathcal{Y}_i \neq \mathcal{Y}_j, \quad \mathcal{I}_i \neq \mathcal{I}_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{4}$$

The study and application of FTL have garnered attention, with 36 distinct research projects focusing on this area, as shown in Figure 7. Notably, FedHealth, an FTL framework for wearable healthcare systems proposed by Chen et al. [5], enables the aggregation of personalized models without imposing restrictions on the structure of smartphone-based human activity recognition data. In addition, Guo et al. [24] presented a scalable FTL framework for Wi-Fi Indoor Positioning, which utilizes Channel State Information (CSI). The efficacy of this framework was evaluated in three different indoor environments of varying sizes to further explore FTL potential.

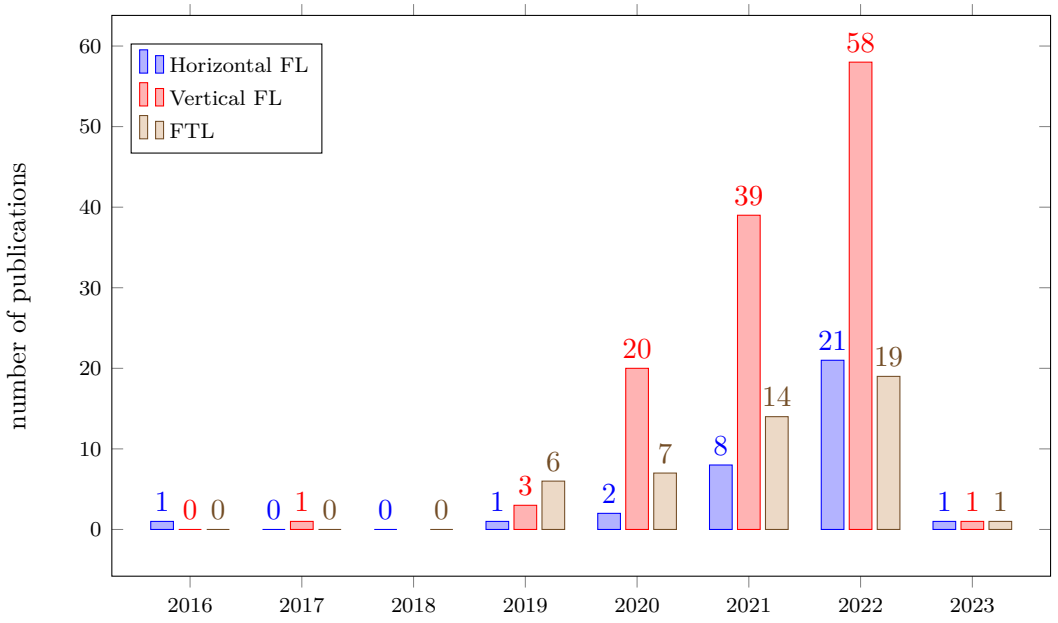

Figure 7: Number of FL' data partition researches from 2016 to January 2023 based on **dblp** (Computer science bibliography).

Comparison

The progression of FL categories from HFL to VFL, and ultimately to FTL, has been guided by both their advantages and limitations, as shown in Figure 7 and Table 3. HFL was at first popular because it trains a common global model and has an easy deployment method. One significant problem is that it fails in real-world situations when the data dimensions of the entities differ. In response, although it needs more complex computing,

VFL, which allows collaborative feature learning, has gained popularity, particularly in the financial industry and recommendation systems. FTL, which enables the collaborative transfer of knowledge between various activities, was launched in 2019 to further counteract HFL's constraint. FTL is adaptable in many practical settings despite the possibility of negative transfer, since it maintains diverse data characteristics among entities having a similar type. As a result, the unique system requirements have a significant impact on the choice of FL data partitioning category, underscoring their significance in the decision-making.

Category	Reference	Domain	Contribution	Advantage	Disadvantage
Horizontal FL	Gao et al.[20]	Medical	Hierarchical Heterogeneous Horizontal Federated Learning for Electroencephalography	Train a shared global model	Not suitable when organizations have different feature set
	Wei et al. [21]	Networking	The Deep FLow Inspection Framework Based on Horizontal Federated Learning		
Vertical FL	Zheng et al. [6]	Finance	A Vertical Federated Learning Method for Interpretable Scorecard and Its Application in Credit Scoring	Learn joint features	More intensive Computation
	Efe [23]	Finance	A Vertical Federated Learning Method For Multi-Institutional Credit Scoring: MICS		
Transfer FL	Chen et al. [5]	Medical	A Federated Transfer Wearable Healthcare	Transfer knowledge between different tasks	Risk of negative transfer
	Guo et al. [24]	Networking	A Federated Transfer Learning Framework for CSI-Based Wi-Fi Indoor Positioning		

Table 3: Examples of related works that focus on FL data partition's category.

4.2.2. Degree of federated Learning

The nature of FL can be differentiated into "Cross-Silo" or "Cross-Device", contingent upon the dimensions and robustness of the training cohort, along with the count of devices in play. This classification is particularly in the purview of participating entities.

- **Cross-Silo Federated Learning**

  Cross-Silo FL becomes relevant when there is a limited, albeit sizeable range (for instance, anywhere between two to a hundred) of participating entities like corporations or institutions (such as hospitals and schools) that engage in learning by training a model on their local data. As per Huang et al. [25], the prime challenges of Cross-Silo FL encapsulate: Firstly, the effectiveness and efficiency to ensure rapidly developed models meet the client's satisfaction and are delivered at a low cost, despite client heterogeneity. Subsequently, privacy and security concerns arise, as data privacy needs to be maintained and potential malicious adversaries detected. Lastly, encouraging cooperation and fostering incentives is crucial to facilitate collaboration amongst clients.

- **Cross-Devices Federated Learning**

  This configuration involves a multitude of small, geographically distributed devices, such as smartphones, smartwatches, and edge devices. The quantity of these devices can reach up to millions, each possessing a comparatively small amount of data and lower computational capabilities relative to the cross-silo FL paradigm. Yang et al. [26] identifies several challenges associated with this structure. Firstly, potential privacy leaks pose significant concerns. These may result from various attacks, such as model extraction, model inversion, and membership inference attacks, potentially leading to the unauthorized exposure of model parameters or training data. Secondly, the limited computing power of participating devices can pose challenges in the implementation of effective FL. Lastly, issues pertaining to incentives and fairness must be addressed. Ensuring equitable participation can be accomplished through the adoption of innovative mechanisms that rely on technologies like game theory and blockchain [27].

Comparison

Table 4.2.2 provides an insightful comparison between two prominent types of FL degrees: Cross-Silo and Cross-Device. Cross-Silo FL usually involves a smaller number of large entities, each possessing a large volume of data and substantial computational resources. However, cooperation between these entities is crucial and can pose a challenge due to the necessity of robust privacy measures and incentives. Conversely, Cross-Device FL involves a vast number of small devices such as smartphones, each holding a small amount of data and lower computational resources. This type of learning raises significant

concerns about privacy, as it's highly vulnerable to various types of attacks, and the equitable participation of devices requires careful attention to incentive mechanisms. In terms of use cases, while Cross-Silo learning is suited to scenarios with uniform data dimensions across entities. Cross-Device learning proves useful in situations involving large-scale distributed data sources with limited computational power. Despite their unique features, both learning types face challenges regarding privacy, computational resources, and the creation of effective incentive mechanisms.

Criteria	**Cross-Silo Federated Learning**	**Cross-Device Federated Learning**
Reference	Ganapathy[28]	Yang et al.[26]
Participants	Limited number of large entities (e.g., corporations, institutions)	Massive number of small devices (e.g., smartphones, smartwatches)
Data Ownership	Each entity has a large amount of data	Each device has a small amount of data
Computational Resources	Entities typically have high computational resources	Devices typically have lower computational resources
Privacy & Security	High emphasis on maintaining data privacy and detecting potential adversaries	High risk of privacy leaks from attacks like model extraction, inversion, and membership inference
Cooperation	Need to encourage cooperation and provide incentives for entities	Need to ensure equitable participation and incentive mechanisms
Use-cases	Ideal for scenarios where data dimensions are uniform across entities	Useful in scenarios with massive distributed data sources with limited computational power
Challenges	Client heterogeneity, privacy and security, cooperation incentives	Privacy leaks, limited computational power, fairness and incentives

Table 4: Comparison between Cross-Silo and Cross-Device Federated Learning.

4.2.3. Network Topology in Federated Learning

The structure of an FL network, often referred to as its topology, delineates the manner in which its constituent entities are interconnected and how they interact for the exchange of information. Below, we discuss the most commonly employed topologies within the context of an FL network.

- Centralized FL, as illustrated in Figure 8, hinges on a singular server that facilitates the collection of local AI models, their subsequent aggregation, and the propagation of the globally computed model. The network topology in use features a single server, complimented by multiple participant nodes [4].
- The concept of Peer-to-Peer Decentralized FL (P2P DFL) is introduced by Behera et al. [29]. This approach aims to eliminate the requirement of a central server for model aggregation. Instead, it utilizes a peer-to-peer communication framework, allowing machines to exchange information directly among themselves. In the architecture proposed by Behera et al. [29], the P2P structure consists of a leader (Aggregator) node, a set of follower (Participant) nodes, and candidate nodes interested in a leader election. This decentralized approach mitigates potential risks associated with a central server, such as operational failures or security vulnerabilities. The architecture is depicted in Figure 10.

As part of the broader movement to decentralize FL, several other architectural frameworks have been proposed, including ring and hybrid network topologies [28]. The hybrid topology, depicted in Figure 9, groups participants who are sequentially arranged according to their use case. Each group member trains their AI model asynchronously before forwarding it to the adjacent participant. The last participant in the sequence then shares the model with the aggregator. The ring topology, shown in Figure 11, operates similarly to the P2P FL. However, the model aggregation is performed by each participant sequentially.

Comparison

Table 5 shows a comparison of the different FL designs, which tells us that each one has its own set of benefits and drawbacks. Centralized FL has one server that coordinates all the nodes. This gives it a high level of scalability, but it may not be the best option for privacy and fault tolerance. P2P FL, on the other hand, gets rid of the central server. This improves privacy and fault tolerance but comes at the cost of more connection overhead and less scalability. With its unique ring structure connecting nodes, Ring FL offers a balanced solution that minimizes transmission overhead while ensuring a moderate level of fault tolerance, privacy, and scalability. Lastly, because Hybrid FL uses parts of both centralized and decentralized models, the transmission overhead, fault tolerance, privacy, and scalability of each hybrid design are different. So, the choice of architecture rests a lot on the needs and limitations of the system in question.

4.3. Discussion

FL is a multifaceted and expansive field, its relevance and applications extend across numerous real-world scenarios permeated by AI, such as healthcare [20, 5], industrial applications [30, 31], and the financial sector [6, 23]. Yet, to fully harness its potential, it's advisable to first clearly outline the parameters that govern distinct FL approaches. This initial step serves as the foundation for developing a customized framework, methodology, or strategy for FL, which is shaped by the specific category of the approach being employed. For instance, the performance of FL aggregation algorithms in HFL might not be as proficient when applied within the context of FTL. Furthermore, the degree of FL must be clearly stipulated to ascertain the system's complexity level and determine the requisite level of data privacy. It's also important to note that designing an effective architecture necessitates the precise specification of the network architecture. In conclusion, to navigate the intricacies of FL and capitalize on its extensive potential, a thorough understanding of the FL category, degree, and network architecture is essential. These parameters significantly influence the resultant system's performance, complexity, and data privacy.

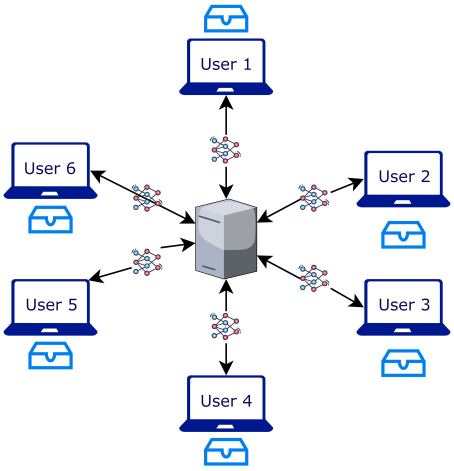

Figure 8: Centralized Federated Learning Architecture.

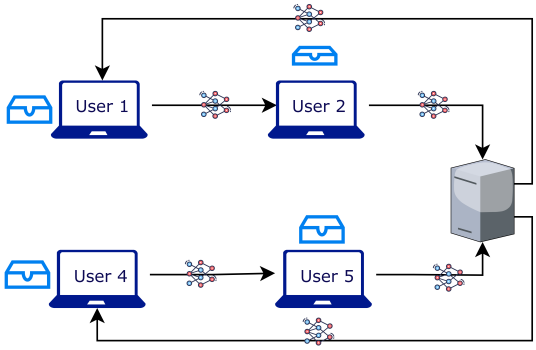

Figure 9: Hybrid Federated Learning Architecture.

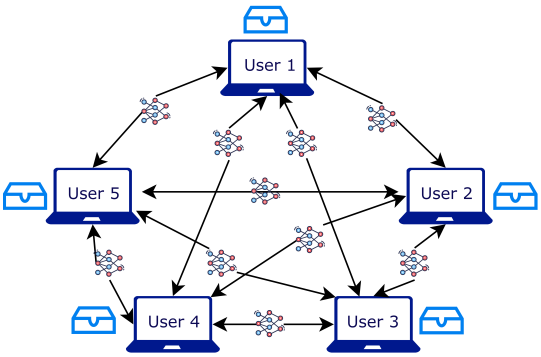

Figure 10: Peer-to-Peer Decentralized Federated Learning Architecture.

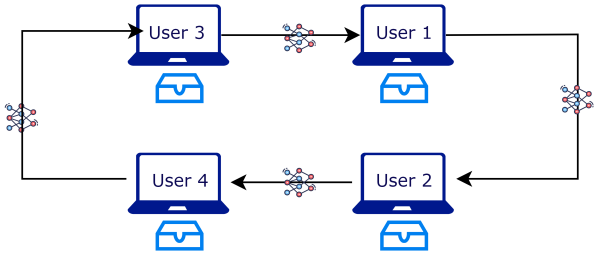

Figure 11: Ring Federated Learning Architecture.

Criteria	Centralized	P2P	Ring	Hybrid
Reference	McMahan et al.[4]	Behera et al. [29]	Ganapathy [28]	Ganapathy [28]
Architecture	Single central	Direct communication between nodes without a central server.	Each node communicates with its two nearest neighbors	Combination of centralized and decentralized aspects.
Communication Overhead	Dependent on the number of nodes.	Highest: grows with the square of the number of nodes.	Low; constant with respect to the number of nodes.	Varies; dependent on the specific hybrid design.
Fault Tolerance	Low; if the central server fails, the system fails.	High; system can still function if a node fails.	Moderate; system can still function unless a critical node fails.	Varies; dependent on the specific hybrid design.
Privacy	Low; central server may see all updates.	High; nodes share updates only with their peers.	Moderate; updates are shared only with neighboring nodes.	Varies; dependent on the specific hybrid design.
Scalability	High; easy to add new nodes.	Low; addition of new nodes increases communication complexity.	Moderate; addition of new nodes requires reconfiguring the ring.	Varies; dependent on the specific hybrid design

Table 5: Comparative Analysis of Federated Learning Architectures: Centralized, Peer-to-Peer, Ring, and Hybrid.

5. Federated Learning Challenges

FL, as previously defined, represents a collaborative process that unites a range ofentities, such as edge devices, to achieve a consistent level of learning. This is achieved through the improvement of the latter, despite the significant differences in training, network, and system architecture amongst these entities. It should also be noted that the distribution and the domain of the data acquired by these entities may not be uniform in real-world applications. These distinct conditions introduce unique challenges due to the heterogeneity of the participants [32]. Moreover, even though private data cannot be shared amongst these entities, there is potential for certain entities to engage in malicious activity. This could involve the distribution of false parametric data or the extraction of private data via the use of the same parametric data, leading to significant privacy concerns. The primary challenges associated with FL are depicted in Figure 12 and will be elaborated upon in the following sections.

5.1. Expensive Communication

FL frameworks possess the capacity to manage an enormous number of remote participant devices, including millions of intelligent entities. These entities engage in sharing large-scale AI models, such as neural networks, with the aggregator. These models, which encompass millions of parameters, require frequent updates to reach desired convergence. However, they often grapple with limited network bandwidth. This limitation gives rise to a significant communication cost challenge in FL [71]. It directly influences the efficiency, scalability, and overall performance of the learning process, thus emerging as a crucial area of focus.

To further curtail communication in such a context, several techniques are currently in use. For instance, adaptive communication strategies such as those proposed by Luping et al. [33] have proven effective. They introduced a communication-mitigating FL framework that permits only relevant local updates to be sent to the aggregator. This approach not only expedites convergence but also diminishes the number of communications required, thereby minimizing the number of shared model parameters. Another approach to reducing communication is increasing the number of local epochs, which reduces the number of CR. This strategy has been effectively employed by McMahan et al. [4]. One can also shrink the size of the messages transmitted in each round by compressing them. For example, Zhu et al. [34] proposed a model compression and privacy-preserving framework for FL. This framework condenses local models by eliminating redundant data and introducing a layer of noise. Similarly, the Federated Learning with autoencoder compressions (FLAC) framework proposed by Beitollahi and Lu [35] uses autoencoders to compress local models.

Finally, limiting the number of participants via selection techniques can also reduce communication needs. McMahan et al. [4] employed a strategy of a random selection of participants for each CR. Moreover, an adaptively partial model aggregation strategy in

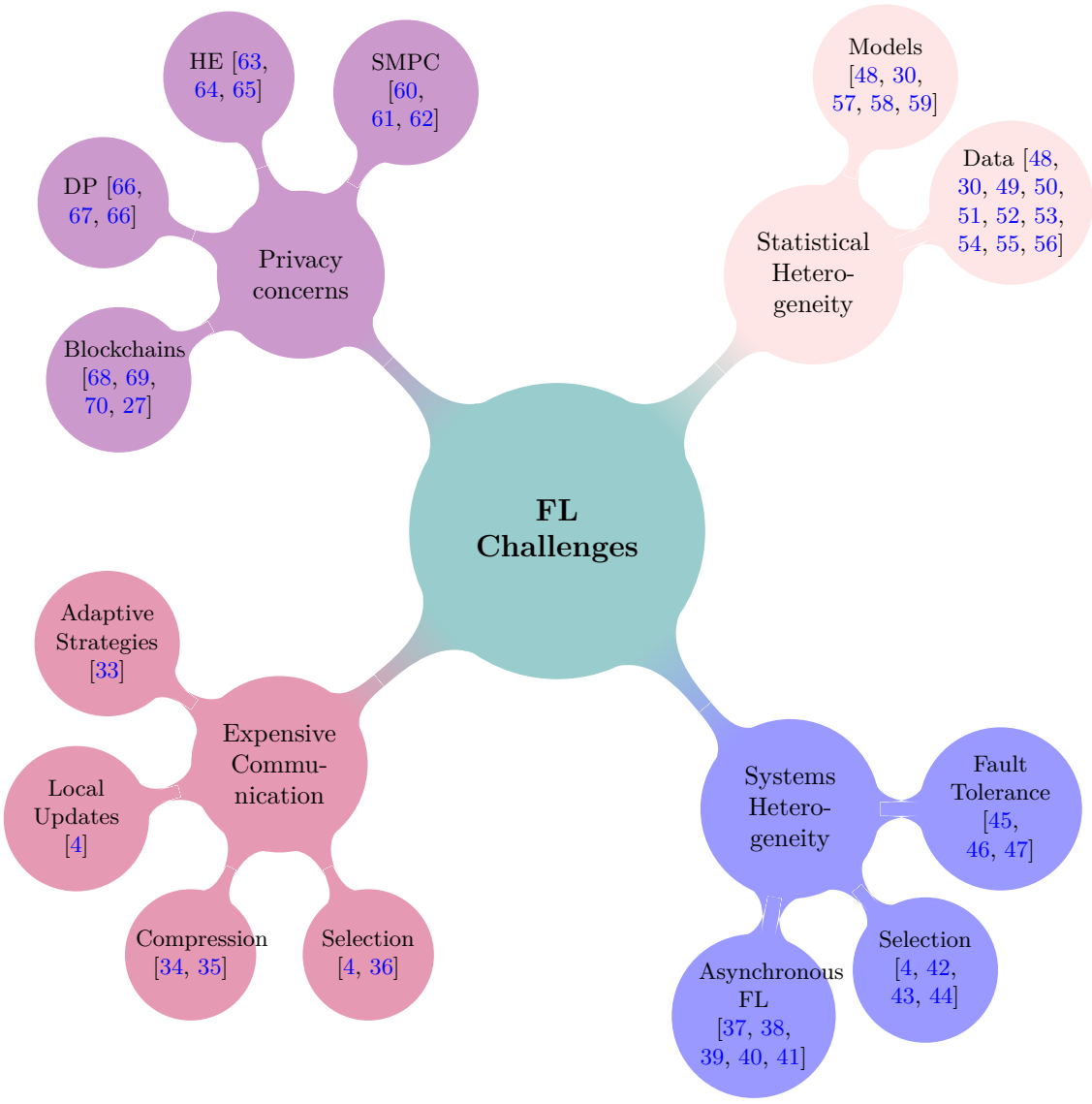

Figure 12: FL Challenges.

FL using reinforcement learning has been proposed by Liu et al. [36] to optimally select the number of devices involved. In their work, Wang et al. 2022 have pointed out that partial customer participation in FL could potentially lead to objective inconsistency, resulting in slowed convergence. Furthermore, they proposed a strategy targeting the enhancement of convergence analysis by advocating for an optimal and unbiased sampling method.

5.2. Heterogeneity in Systems

System heterogeneity is inherently induced by discrepancies in hardware attributes such as CPU, GPU, and RAM [32], variations in network connectivity (e.g., 3G, 4G, 5G, WiFi) [32], and fluctuations in power capacity (battery level) [32]. This diversity may give rise to statistical heterogeneity [73].

FL networks can be particularly expansive, with often only a minuscule fraction of the devices being active at any specific moment [32]. Edge devices may exhibit unpredictability, prone to dropping out due to constraints in connectivity or power [32]. This implies that any participant's device could potentially withdraw from the collaboration due to network glitches or battery limitations [32].

Several strategies have been put forth by researchers to mitigate these challenges. Such strategies encompass asynchronous FL communication [37, 38, 39, 40, 41], strategic client selection [4, 42, 43, 44], and the implementation of fault tolerance mechanisms [45, 46, 47]. Each of these solutions will be further elucidated in the ensuing sections.

5.2.1. Communication in Asynchronous Federated Learning

FL frequently aggregates models from participant devices through synchronous communication, as depicted in Figure 13-a. However, this method is hindered by constraints on participant device computation and transmission bandwidth, often leading to disruption in learning. An alternative approach involves asynchronous communication, illustrated in Figure 13-b. In this approach, participant devices perform local training asynchronously; local AI models are transmitted to the aggregator server at varied times. This can mitigate efficiency bottlenecks caused by participant devices lagging in synchronous communication [37].

The adaptive nature of asynchronous FL in wireless networks has been explored by Lee and Lee [38]. They proposed an adaptive transmission planning method considering variations in wireless channel quality. Another noteworthy work by Xie et al. [40] introduced an asynchronous FL aggregation method that updates the global AI model upon receiving any participant model with a low degree of staleness. This method was further employed by Sprague et al. [39] for geospatial applications. Lastly, Chen et al. [41] proposed the ASO-Fed framework, an asynchronous FL model under a non-IID data settings. It enables clients to perform online learning with continuously incoming dynamic data. This framework uses a regularization and central feature learning module to expedite the learning process on how clients relate to each other.

5.2.2. Client Selection

Client selection seeks to enhance the efficiency of FL by meticulously choosing entities that neither cause delay nor disrupt the aggregation phase of models due to resource insufficiency. Furthermore, it ensures these entities do not trigger system failure due to their own malfunction. McMahan et al. [4] proposed the random selection of participants for each FL CR. However, this approach may inadvertently include participants with limited resources

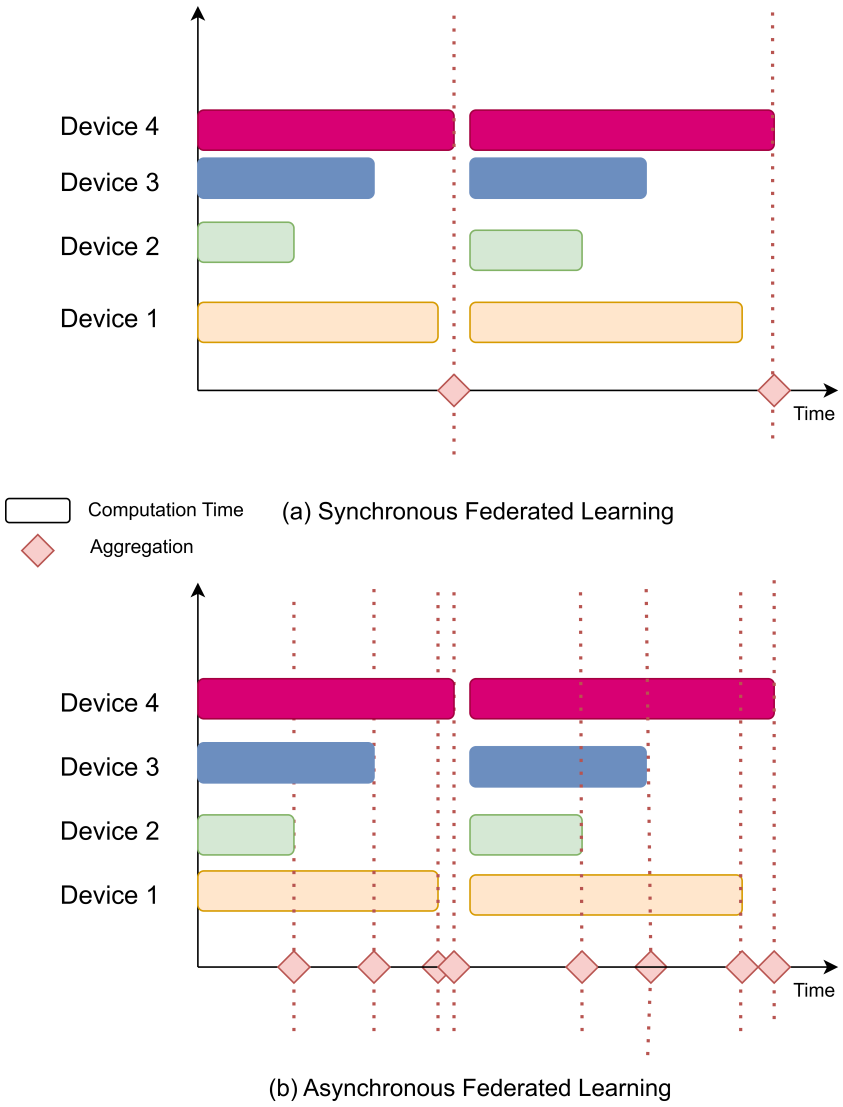

Figure 13: Federated Learning Communication Type.

or scarce data. Contrarily, Nishio and Yonetani [42] presented the Federated Client Selection (FedCS) framework for FL, designed to select participants who demonstrate minimal AI model update and upload times. This selection is subject to the constraints of computing and connection resources. In FedCS, the server actively solicits resource details from all participating nodes, then formulates the issue akin to a classic optimization problem, specifically, the Knapsack Problem. Here, the maximum time of an FL round is equated to the maximum weight of the knapsack, and a greedy algorithm is deployed to reduce the

complexity of the problem.

In a bid to augment energy-constrained training, Wu et al. [43] introduced the FL Energy (FedLE) framework. This framework forms clusters of participants using cosine similarity metrics of all local AI models, assigning a higher probability to clusters with fewer clients and a lower probability to those with more clients. Within each cluster, participants with a larger battery capacity are given preference to optimize energy usage. Furthermore, Wehbi et al. [44] proposed the Federated Minimum Interference (FedMint) framework. This two-sided client selection mechanism chooses clients based on their own preferences as well as those of the aggregators. Preferences include aggregator rewards for clients and model accuracy for aggregators, providing incentives for clients to maximize their computational and communication resources. The framework also assigns an initial accuracy value to any new IoT device, thereby allowing it to participate in the collaborative process.

5.2.3. Fault Tolerance

Fault tolerance in FL represents an evolving domain aimed at maintaining uninterrupted learning processes despite hardware or software malfunctions [47]. Several strategies have been deployed to ensure fault tolerance within this milieu, such as:

- **Checkpointing**: This strategy involves periodic backup of the system state to facilitate recovery, particularly effective when communication processing time is protracted [46].
- **Heartbeat Monitoring**: This method ensures regular transmission of messages across various system components to verify their functional status [46].
- **Redundant Processing**: This technique encompasses performing identical calculations on multiple devices or nodes, thereby enhancing system reliability and fault tolerance [74].
- **Error Detection and Correction**: Implementing mechanisms to identify and rectify errors in model updates or other system components helps maintain system integrity and performance. Various tools such as checksums, error-correcting codes, formal methods, and consensus methods can be employed. For instance, the Byzantine-resilient variant of the Stochastic Gradient Descent algorithm [39] has been proposed to enhance resilience and error detection and correction capabilities in a distributed learning environment.
- **Robust Aggregation Methods**: These strategies are resilient to noise or malicious models and contribute to the overall stability of the system [75].

Statistical heterogeneity presents a significant obstacle to the successful execution of FL [32]. It introduces complexity to the aggregation process and instigates discrepancies

between local and global AI models. The root of this issue lies in the potential variance in both size and distribution of local datasets employed for training participant AI models. This variation can lead to performance disparities amongst local models and pose challenges to their successful amalgamation into an effective global model.

Even in scenarios where dataset characteristics do not significantly influence FL, the uniqueness of AI models per participant cannot be overlooked. Despite being of the same type (e.g., CNN local and global models), these AI models may differ in architecture and parameters, such as the number of layers and parameter These dual layers of complexity stemming from statistical heterogeneity will be explored in further detail subsequently..

5.2.4. Data Heterogeneity

In FL, data heterogeneity is understood as the distinct variation in the statistical distribution of data across different participants [76]. This condition manifests in the form of non-identical and independently distributed (non-IID) datasets belonging to the participants [77]. This discrepancy implies that the datasets may differ in terms of size and distribution, subsequently leading to variations in the performance of local models. As an illustrative case, consider two electronic healthcare record devices containing disparate and unbalanced datasets [78].

Another manifestation of data heterogeneity in FL is domain shift [79], where local datasets possess various characteristics, features, and balance. Such disparities can considerably impact the performance and convergence of the AI models trained by the participants [80]. A practical example can be seen with two industrial robots, R1 and R2, located in China and France, respectively. Even though they employ the same CNN model for FL, they may acquire distinct types of image data for a face recognition problem via different IoT devices, thus creating a domain shift.

Recent research has primarily focused on addressing these categories of challenges. For instance, personalized FL and TL have been used to tackle this issue [48, 30], while clustering techniques have been implemented to group similar participants according to data [49, 50, 51]. Additionally, normalization of local DL models has been attempted [52], as well as the deployment of a public dataset that includes groupings constituting a fraction of local datasets [53]. Domain adaptation techniques have also been employed to resolve the domain shift problem [54, 55, 56]. These research efforts will be elaborated upon in Section 6.

5.2.5. Model Heterogeneity

Traditional FL aggregation algorithms usually demand that all participants employ an identical AI model architecture [4]. However, this norm may be impracticable in certain real-world settings [81]. Primarily, the number of IoT sensors can differ across various scenarios, leading to a range of datasets and their corresponding AI models. In addition, there could be instances where a participant's device lacks the necessary computational capacity to train a more intensive model, effectively barring them from participation. Consequently,

model heterogeneity arises when the devices participating in the learning process employ divergent AI models, or models that possess distinct features and characteristics.

A substantial body of FL research is geared toward addressing this challenge. For example, [48, 30] have suggested algorithms that aggregate either personalized or foundational AI models. Moreover, innovative frameworks like FedMD [57], FedH2L [58], and MHAT [59] have been put forth, leveraging knowledge distillation [82] as a method to exchange and aggregate predictive outputs rather than local AI model parameters. These studies and their significant contributions will be discussed in more detail in Section 6.

5.3. Privacy Concerns

Ensuring data privacy and transaction security are paramount issues in FL applications [63]. Traditional centralized learning raised concerns over the secure sharing of user data. Although FL ameliorates this problem by keeping user data local, it introduces a new challenge: ensuring the security of shared model information, such as weights or gradients. In this context, several privacy risks are still to be addressed, including:

- **Data Poisoning:** This threat [83, 84] arises when a participant intentionally or inadvertently transmits incorrect data to the aggregator, which may negatively impact the model's accuracy.
- **Data Leakage Attack:** In this scenario [85, 86, 87], an attacker intercepts the model's weights to reconstruct the original data. Studies, such as those by [88], highlight the vulnerability of FL to gradient data leakage, thereby compromising the privacy of participants' training data.
- **Model Inversion Attacks:** These attacks [89, 90] occur when an adversary leverages the updates provided to the aggregator to reconstruct an approximation of the original model.
- **Membership Inference Attacks:** Here, an adversary [91, 92, 93] utilizes the trained model to deduce the participation of specific members in the FL process, potentially revealing sensitive information about the data owners.

Recent research focuses on augmenting the privacy of FL using diverse tools such as Secure Multi-party Computation (SMPC) [94], Homomorphic Encryption (HE) [95], and Differential Privacy (DP) [96, 97], as highlighted in [32]. These strategies will be elaborated upon in the following sections.

5.3.1. Secure Multi-party Computation (SMPC)

SMPC, a subfield of cryptography [94], strives to devise novel methodologies that facilitate distributed entities to collaboratively compute a function over their inputs while preserving the confidentiality of these inputs. This is typically achieved by distributing the

computation across small tasks, each performed individually by an entity, and subsequently to merge the results. Within the realm of FL, each participant trains the model on their private data utilizing SMPC techniques, such as secret sharing [60], to encapsulate local model updates (either gradients or weights) as shared secrets. Subsequently, aggregation can be performed using HE (discussed in Section 5.3.2) or secret sharing-based protocols [98].

As stated by [99], SMPC delivers robust privacy assurances and dependable protection against adversarial assaults in FL. However, the employment of SMPC is not without its challenges:

- **Computational Complexity.** SMPC protocols often necessitate extensive computation resources, potentially resulting in increased training times and resource consumption.
- **Communication Overhead:** Secure protocols may require increased Communication Rate (CR) and larger messages, potentially leading to higher bandwidth usage and potential communication congestion.
- **Scalability:** Scaling FL systems to accommodate a large number of nodes can be challenging, as the effectiveness of SMPC protocols may diminish as participant count increases.

Regardless of these challenges, contemporary research on SMPC and FL is focused on developing more efficient and scalable protocols that can be adapted to a variety of applications, particularly those with rigorous privacy and security requirements. For instance, Bonawitz et al. introduced an SMPC protocol in FL for secure aggregation that ensures robust privacy [60]. SecureML [61] and FALCON [62] are two exemplary SMPC-based frameworks that maintain the privacy of the training process in ML.

5.3.2. Homomorphic Encryption

HE is a unique class of encryption algorithms, specifically designed to allow the execution of mathematical operations on encrypted data. Fully HE (FHE) [100], is a type of HE that facilitates any form of ciphertext computation. Conversely, Semi-HE (SHE) [101] enables selected ciphertext computations, such as addition and multiplication.

In FL, HE plays a critical role by aggregating model updates across various devices while safeguarding user data privacy. A detailed breakdown of the formal application of HE in FL involves several key components:

1. *Encryption:* This phase involves the encryption of local AI model updates on each participating device. Assume w_i is the model update from a specific device i, and $Enc(w_i)$ is the resulting ciphertext derived from encrypting w_i using a HE scheme. Therefore:

$$Enc(w_i) = HE.Encrypt(w_i) \tag{5}$$

Here, HE denotes the HE function. After this process, the aggregator receives ciphertexts, maintaining their confidentiality.

2. *Aggregation:* Let $Enc(w_1), Enc(w_2), \cdots, Enc(w_n)$ be the set of encrypted model updates originating from n devices. The central server can execute homomorphic operations (for instance, addition or multiplication) on these ciphertexts to achieve the newly updated model:

$$Enc(W_{new}) = W_{old} \otimes Enc\left(\sum_{i=1}^{n} w_i\right) \tag{6}$$

Where $\otimes$ represents the chosen homomorphic operation, and $Enc(W_{new})$ is the encrypted global model.

3. *Decryption:* Let $Enc(W_{new})$ be the aggregate ciphertext procured by combining the encrypted model updates from all devices. Using a secret key, the central server can decrypt this aggregate ciphertext to reveal the final model update.

$$W_{new} = HE.Decrypt(Enc(W_{new})) \tag{7}$$

Here, $HE.Decrypt$ refers to the homomorphic decryption function.

Several studies, such as [63] and [64], demonstrate how HE can mitigate privacy challenges in FL and facilitate model aggregation from diverse devices. For example, the proposed secure FL framework for medical data by these studies strengthens data privacy in a network of hospitals using HE. Other contributions, such as [65], have further improved HE in FL by proposing an efficient doubly homomorphic secure aggregation scheme for cross-silo FL, employing multi-key HE and seed homomorphic pseudo-random generator as cryptographic primitives.

Despite its potential for preserving model training and assessment data privacy in FL, HE does have some limitations:

- The computational complexity of HE is high, leading to increased training time and resource consumption at the node level.
- Some HE systems only offer basic operations like addition and multiplication, which might not suffice for complex ML models or optimization algorithms.
- Compared to traditional encryption methods, HE produces larger ciphertexts, thereby raising communication costs and potentially exacerbating bandwidth constraints.

5.3.3. Differential Privacy

DP [96, 97] serves as a robust methodology for safeguarding the privacy of data during the analysis or dissemination of sizeable and sensitive datasets. The central premise relies on integrating a random perturbation function to the original data, which helps mask

sensitive details without compromising the relevance of the aggregated data [102]. In the context of FL, where participants collaboratively train a model in a decentralized fashion, DP is primarily implemented by infusing random noise into the model parameters. Such techniques may include the Laplace mechanism [103] or the Gaussian mechanism [104], and the noise-infused parameters are then shared with the aggregator [99].

For instance, let's consider a FL system comprising n participants, each labeled as participant i, involved in the learning process. Every participant computes its local model update, denoted as Δw_i, using its unique dataset D_i. To achieve ϵ-DP, each participant i integrates Laplace noise to its local update:

$$\Delta w_i^{DP} = \Delta w_i + Lap(\Delta f/\epsilon) \tag{8}$$

In the above equation, Δw_i^{DP} symbolizes the DP model update, Δw_i indicates the original model update, and $Lap(\Delta f/\epsilon)$ denotes the noise added to attain ϵ-DP, where ϵ is a positive real number that determines the quantity of the noise. The sensitivity parameter Δf is contingent on the specific ML task and the dataset in use. Its careful determination is crucial to ensure the desired level of privacy protection. Post-computation of DP model updates, they are aggregated by the central server to update the global model. This aggregation could be achieved by simply averaging the noisy updates:

$$W_{new} = W_{old} + \frac{1}{n} \times \sum_{i=1}^{n} (\Delta w_i^{DP}) \tag{9}$$

Here, W_{old} denotes the current global model, and W_{new} signifies the updated global model.

Global Differential Privacy (GDP) and Local Differential Privacy (LDP) are two possible approaches to utilize within FL. Firstly, GDP necessitates a trusted aggregator, which incorporates the noise post-aggregation. The majority of FL security research focuses on LDP as it empowers data owners to privatize their data prior to sharing. For example, [66] introduced a novel LDP for FL (LDPFL) protocol designed for industrial settings. This protocol operates effectively with untrusted entities while providing stronger privacy assurances compared to existing methods. Similarly, [67] addressed the issue of privacy within FL and proposed a solution based on LDP to secure user information. Moreover, [66] suggested a framework leveraging LDP for FL, specifically for image classification applications, thereby showing that LDP can enhance privacy while maintaining performance.

The integration of DP within FL can effectively bolster privacy protection for participating devices and their data. However, the employment of DP within FL presents some challenges:

- Utility-Privacy Trade-off: The introduction of noise to ensure privacy can potentially delay convergence and escalate the variance in model updates. The pursuit of an optimal balance between privacy and utility thus emerges as a critical challenge when designing privacy-preserving FL systems.

- Parameter Tuning: The task of adjusting DP noise settings and privacy budgets can be quite challenging. It requires careful consideration of the FL application, data sensitivity, and the level of privacy desired.

5.3.4. Blockchain

Blockchain technology, as discussed by [105], provides an immutable, verifiable, and secure decentralized ledger that records transactions among parties in sequential blocks. When integrated with FL, blockchain can serve as a trustless infrastructure, offering a decentralized framework for secure model training, validation, and updates [106]. Several potential applications of blockchain in FL include:

- **Decentralized model training and updates:** FL can harness the distributed nature of blockchain to facilitate model training and updates without reliance on a central server. Blockchain consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), or Byzantine fault-tolerant techniques can assist participants in reaching a consensus on the updated model.
- **Secure and transparent record-keeping:** Blockchain can serve as a secure repository for storing model updates, training history, and validation results. This capability enhances the audibility and verifiability of the learning process for both participants and auditors.
- **Smart contracts for incentive mechanisms:** In FL, incentivization is essential to ensure nodes provide high-quality data and model updates. Blockchain-based smart contracts can reward users with tokens or improved models in recognition of their valuable contributions to learning.
- **Privacy preservation:** Combining blockchain technology with privacy-preserving methods such as Secure Multi-Party Computation (SMPC) (Section 5.3.1), HE (Section 5.3.2), and DP (Section 5.3.3) can enable secure and private model training in FL. These techniques allow blockchain nodes to share encrypted data or model updates without compromising their private information.
- **Secure model sharing and access control:** Blockchain can securely distribute trained models among participants or external parties. It achieves this by establishing access control policies through smart contracts, limiting model access strictly to authorized parties.
- **Model provenance and intellectual property protection:** Blockchain's ability to track trained model provenance and ownership aids in protecting intellectual property and verifying the authenticity of models deployed in various applications.

The prospect of combining blockchain and FL is generating considerable interest. Numerous research studies have examined the potential benefits and challenges of integrating

these two technologies. For instance, Kim et al. [68] conducted an investigation of potential security vulnerabilities in blockchain-enabled FL, proposing effective solutions to these challenges. Additionally, Hardjono and Pentland [69] explored how blockchain technology could facilitate identity and access management in FL, emphasizing the importance of security and interoperability. Chen et al. [70] presented a secure and robust IoT-enabled smart city framework that blends distributed ML and blockchain technology, with a particular focus on FL for edge devices. Lu et al. [27] developed a framework that ensures privacy in Industrial IoT data sharing by combining FL and blockchain, thereby promoting safe and private data sharing and analysis.

Despite the potential of blockchain technology to address many inherent challenges in distributed ML environments including those related to security, transparency, and decentralization it is important to consider potential trade-offs and obstacles. These could encompass the elevated computational expense, scalability issues associated with blockchain consensus methods, and the intricacy of integrating blockchain within established ML frameworks. Nevertheless, when appropriately integrated with FL, blockchain can significantly enhance the overall efficiency and efficacy of the system.

5.4. Discussion

Numerous existing solutions have been presented to address associated challenges, including the enhancement of communication efficiency, the implementation of synchronization strategies for participant devices, the fortification of system robustness, the design of algorithms for data and model heterogeneity, and the incorporation of privacy-preserving techniques. These latter include SMPC, DP, HE, and blockchain technologies.

Upon analysis, we observe that, while these proposed techniques exhibit a general application, they do not necessarily cater to specific use cases. Furthermore, their solutions may give rise to challenges in different categories. For instance, the use of blockchain technology as a decentralized security method, despite its advantages, can introduce the challenge of increased communication costs during the exchange of model parameters. Therefore, to devise a solution that is both reliable and valuable, all FL challenges must be comprehensively considered.

6. Federated Learning Aggregation Algorithms

FL's primary goal is to assist a group of K participants in making better decisions by minimizing the loss function F_k of their local neural network models, which are built with local weights w_k and trained with private datasets D_k that contain n_k samples. The mathematical formula of the local objective is:

$$min_{w_k \in \mathbb{R}^d} F_k(w_k) \quad where \qquad F_k = \frac{1}{|D_k|} \sum_{i=1}^{n_k} f_i(w_k) \tag{10}$$

To collaborate learning and make enhancement in decision-making, FL works by having a server that collects local model weights for each CR t, aggregates them to create a global neural network model with global weights w^t, and then broadcasts the latter to all participants to make local updates. McMahan et al. [107] started FL in 2016 with **FedSGD**, which is an abbreviation for Federated Stochastic Gradient Descent and is the name of the basic algorithm that makes the aggregation step. With **FedSGD**, first, the participants train their local models using batch GD with just one local epoch and upload their gradient g_k. The central server then collects these latter, averages them, and aggregates them by updating the global weights using the GD algorithm. **FedSGD** requires participants to train their models with one local epoch and one batch, which slows down their convergence and requires hundreds of CR to achieve the target accuracy of all of them.

While various solutions have been proposed to address the challenges of the FL (refer to Section 5), it is noteworthy that certain aggregation methods have emerged to tackle multiple challenges simultaneously during the aggregation process. These methods are designed to target a specific challenge as the primary focus while minimizing any adverse effects on other challenges. By strategically considering the interplay between challenges, these aggregation methods offer approaches to mitigate multiple obstacles concurrently, thus showcasing their potential to enhance the overall performance and effectiveness of FL systems.

The **FedPer** method, proposed by Arivazhagan et al. (2019)[48], employs TL techniques and distinguishes between base and personalized layers in a local model. In the context of CNNs models, these base and personalized layers typically correspond to feature extractor layers and fully connected layers, respectively, such that participants share only the base layers with the server while retaining personalized layers. This approach was tested on the FLICKR-AES and CIFAR datasets, showing superior performance to **FedVG** in personalization tasks. This suggests that using base and personalized layers can alleviate issues from statistical heterogeneity in FL. Contrary to the typical approach, Ek et al. [108] suggested a role reversal for the two layers in FL models. They propose that the base layers should concentrate on individual-specific decision-making, whereas the personalized layers should focus on shared representation learning. Hence, the choice of assigning roles to these two layers is contingent upon the specific problem that needs to be resolved. Chen et al. (2018)[109] proposed a federated meta-learning framework, **FedMeta**, where only models gradients are shared. They used the algorithm of model agnostic meta-learning (MAML) Finn et al. [110] that treats each client as a task. The goal of using meta-learning is to learn a model on a collection of tasks (clients' tasks), such that it can solve new tasks with only a few samples. **FedMeta** executes on two steps per FL CR. Inner update, when each client trains the local model with its training dataset using the weights vectors of the global model, tests its performance with its testing dataset using the updated weights vector, and returns the gradients to the global server. Outer update, if the global server receives local model parameters, it updates the global model and broadcasts it to participants. They tested **FedMeta** on the LEAF datasets with non-

IID data and the real-world production dataset, showing that it is 2.82–4.33 times more efficient in terms of communication cost, has rapid convergence and improves accuracy by 3.23%-14.84% compared to '**FedAVG**. **FedProx** is a FL framework proposed by Li et al. (2020). It generalizes the **FedAVG** approach, with key modifications to address data and system heterogeneity. In **FedProx**, participants optimize the loss function with a proximal regularization term. This term penalizes large divergence between the current local model and the previous global model, aiming to keep models more coherent across the network. In order to address system heterogeneity, they suggest solving the previous objective function directly, rather than training local models over several local epochs. This approach can help decrease computational overhead. **FedProx** was evaluated using Multinomial Logistic Regression (MLR) and LSTM models trained on LEAF and SHAKESPEARE non-IID datasets, respectively. To mimic system heterogeneity, devices were allocated different amounts of local workload. When compared to **FedAVG**, **FedProx** showed substantial improvements, with accuracy increases averaging 22%. This suggests that the framework's considerations for data and system heterogeneity are effective in enhancing model performance in an FL setting. Wang et al. (2020)[111] proposed Federated Matched Averaging (**FedMA**), a novel layer-wise FL algorithm tailored for CNNs and LSTMs models. In **FedMA**, each layer of the model is trained independently and then communicated to the server. The server conducts a one-layer matching process on the participants' first layer weights to derive the weights for the first layer of the federated model. These weights are then broadcast back to the participants, who use them to train all subsequent layers on their local datasets, while keeping the federated layer frozen. This process is iteratively repeated until all layers have been processed. The number of CR in **FedMA** corresponds to the number of layers in the local models for all participants. The algorithm was evaluated using a VGG-9 model trained on the CIFAR-10 dataset, and an LSTM trained on the Shakespeare dataset. The results suggest that **FedMA** does not result in communication overload between the server and participants, as it only shares one layer at a time, unlike **FedAVG** and **FedProx** that share the entire models. Consequently, **FedMA** potentially offers a more scalable and efficient approach for FL.

The **FedDist** technique, proposed by Ek et al.(2021)[51], is a novel FL aggregation method developed for Human Activity Recognition (HAR). This method modifies the model architecture by identifying variations among individual neurons across different clients to address divergence in heterogeneous and non-IID datasets. Initially, **FedDist** aggregates local models using the standard FedAvg approach. It then measures the pairwise dissimilarity between each neuron in a client's layer and the corresponding neuron in the global model. If the distance exceeds a predefined threshold, that neuron is incorporated into the corresponding layer of the global model. Following this, clients perform layer-wise training: they freeze the updated layer and all layers above it, and train the subsequent layer. This iterative process yields new global and local models with adjusted neuron counts per layer. They evaluated **FedDist** by training a Convolutional Neural Network (CNN) model on the LEAF dataset. The results showed that **FedDist** outperformed

FedAVG, **FedPer**, and **FedMA**, demonstrating its potential for handling non-IID and heterogeneous data. However, one drawback of **FedDist** is the high communication cost involved. By aggregating layers and adding extra neurons to the global and local models in each FL CR, **FedDist** significantly increases the communication load between clients and the global server.

Briggs et al. (2020) [49] proposed a combination of FL and Hierarchical Clustering (**FL+HC**) to improve learning on non-IID data. This approach groups similar participants' local models, aiming to mitigate their divergence. The initial aggregation is performed using **FedAVG** [4], followed by the execution of the Agglomerative Hierarchical Clustering algorithm [112] to select clusters of similar local models. Finally, an additional aggregation step is performed at each cluster level in parallel. The authors evaluated the performance of **FL+HC** on an image classification task, using a CNN model trained on the FEMNIST dataset in both IID and non-IID settings. They compared the results with a standalone FL approach. The experiments demonstrated that **FL+HC** allows for quicker learning, even with variations in the algorithm's hyperparameters. However, there are some caveats to consider. The **FL+HC** approach assumes homogeneity of local models, limiting its applicability in scenarios with diverse model structures. Additionally, in the second step involving cluster aggregation, having multiple aggregation algorithms running in parallel on a central server may introduce issues related to synchronization and reliability.

FedGA, proposed by Guendouzi et al. (2022) [30], is an aggregation algorithm that combines the concepts of Federated Personalization (FedPer) and Genetic Algorithms (GA). This approach is aimed at addressing the challenges associated with data and model heterogeneity in federated learning environments. In the initial setup, each participant shares a personalized fraction of their dataset with the central server, and all participants have the same base layer structure for their local models. In the **FedGA** approach, only the base layer weights of each local model are transmitted to the central server. The server then employs a genetic algorithm to compute the new global weights by seeking the individual (i.e., set of weights) that minimizes the loss function of the global model, based on the union of the data subsets shared by the participants. The authors evaluated the proposed approach on the MNIST dataset, using both IID and non-IID settings and employing CNNs models for local and global learning tasks. The results showed faster convergence of all local models and improved average accuracy when using **FedGA** compared to **FedPer**, thereby reducing communication costs. However, it's worth noting that the use of genetic algorithms may increase the computational complexity of the aggregation process compared to simpler methods, such as directly averaging the weight vectors.

Berghout et al. (2022) [52] proposed a heterogeneous FTL approach named **FTL-RLS**. This approach is particularly adept at handling the challenges of statistical heterogeneity and system heterogeneity in FL, in addition to offering remarkable aggregation speed and reducing communication costs. The **FTL-RLS** employs the Recursive Least Squares (RLS) algorithm to enable rapid computation at each client's end involving a single-weight matrix. Prior to the aggregation step, **FTL-RLS** encodes the weight matrices and standardizes

their dimensions using a transpose operation. This tackles issues related to diverse model architectures and data privacy. During the aggregation phase, **FTL-RLS** normalizes the learning weights by using the standard deviation and mean values of each matrix to cater to non-IID data challenges associated with disparate weight scales. Post-aggregation, linear algebra operations are performed to reverse the prior steps. In their experiments, Berghout et al. used a personalized dataset distributed across four clients. The results indicated that **FTL-RLS** achieves an impressive execution time and reduces the communication message size. However, in terms of model accuracy, **FTL-RLS** performs on par with other methods. It's worth noting that, due to its reliance on RLS methods, **FTL-RLS** is primarily suitable for regression tasks.

Ye et al. (2022) [113] proposed an anchor-based feature matching aggregation method for FL, termed as **FedFM**. This approach effectively addresses the challenge of data heterogeneity across clients by rectifying inconsistencies and overlaps in features and facilitating local model convergence. They introduce the concept of anchors, markers used to reduce the distance between the central anchor of a particular data category and the data within that category. Simultaneously, the anchors increase the distance between a data category and other categories. Consequently, the aggregation process is performed for the anchor of each category along with the global model. To mitigate bandwidth costs between clients and servers, potentially exacerbated by their solution, they propose a communication scheme where both models and anchors are communicated within a single handshake. Moreover, while anchors are communicated in each FL round, models are only communicated during specific rounds. Upon comparing **FedFM** with **FedAVG**,**FedProx**, and other state-of-the-art algorithms using CIFAR and CINIC-10 datasets and RestNet pre-trained models, it was observed that **FedFM** improves the learning accuracy by an average of 6%, while incurring minimal communication time costs.

Palihawadana et al. (2022) [114] proposed **FedSim**, a FL aggregation algorithm that mitigates model divergence due to unequal distributions (non-IID) datasets across participants. The method is based on comparing the gradients of models to determine their similarity. **FedSim** decomposes the baseline aggregation process into local and global aggregation steps. After the initialization and broadcasting of the global model, and the collection of gradients from the local models of the participants, the aggregator reduces the matrix dimension of the collected gradients using Principal Component Analysis (PCA) [115]. It then constructs clusters of participants using the K-means ML algorithm [116], executes the local aggregation for each cluster, averages their results in the global aggregation step, and broadcasts the final global model throughout the network. **FedSim** was evaluated on various datasets, including FedME-x and Fed-Goodreads. The results in terms of global model accuracy confirm the superior of its performance when compared with **FedAVG** and **FedProx**, primarily due to the use of clusters that expedite convergence. However, a noticeable increase in complexity is observed when PCA and K-means are executed for each FL CR.

Li and Wang (2019) [57] proposed **FedMD**, a FL framework that leverages TL and

knowledge distillation [82] to obviate the need for deploying a singular homogeneous AI model. The process starts with the deployment of a public dataset on all the participant devices. These datasets serve as the basis for the initial training of their local models. Following this, the participants release their prediction outputs, which are subsequently averaged by the aggregator. The final averaged predictions are then shared with all the participants, who strive to minimize the discrepancy between their individual predictions and these averaged predictions. Despite evaluating **FedMD** on various datasets, including LEAF and CIFAR (both IID and non-IID), the results do not necessarily exhibit high performance, but they do underscore the potential utility of **FedMD** when using multiple heterogeneous models.

Hu et al. (2021) [59] proposed Model Heterogeneous Aggregation Training **(MHAT)**, an innovative FL model heterogeneous aggregation training scheme. This method addresses challenges associated with heterogeneous model architectures and the communication costs between clients and the server by primarily focusing on model outputs rather than parameters. It integrates FL and TL techniques, specifically knowledge distillation. In the first step, participants train their local models and share the prediction outputs with the aggregator. The aggregator then aggregates these outputs and generates new ones, which are subsequently broadcasted to all clients. These new outputs form part of the training dataset, thus enhancing the learning of all participants' local models without affecting the architecture of these models. **MHAT** was evaluated using the IID MNIST dataset under both homogeneous and heterogeneous models. The results showed a difference of 82 FL communication rounds between the baseline FL (**FedAVG**) and **MHAT** to achieve an average accuracy of 95% across all clients, demonstrating the algorithm's efficiency in reducing the resource consumption for clients.

Li et al. (2021) [58] proposed **FedH2L**, a framework that addresses model and statistical heterogeneity issues in the FL. This is achieved by focusing on the exchange of prediction outputs rather than local model parameters, and it also supports decentralized FL, thereby eliminating the need for a third-party entity. Initially, each participant computes its local gradients using its own private datasets, makes predictions using public datasets, and shares these prediction outputs and their corresponding accuracies with all other participants. These participants then calculate the public gradients, defined as the losses between the predictions of their local models and the public predictions. Subsequently, they update their local models using these new gradients. **FedH2L** was evaluated on the MNIST and Office-Home datasets [117]. The results showed that **FedH2L** outperforms several state-of-the-art algorithms and addresses system and communication resource constraints by reducing the training local epoch to one and communicating only output predictions and accuracies, rather than model parameters. However, this framework requires thousands of FL CRs to achieve a comparable accuracy level, and it only supports homogeneous data, which may not accurately reflect real-world situations. Ahmed et al. (2021) [118] proposed an innovative FTL model that accommodates the varying computational resources and model architectures available to different clients. In their system, participants are

categorized into clusters based on the computational resources they possess, classified as either low, medium, or high. In the initial stage, the server initializes a unique global model for each cluster. Then, the server associated with each cluster independently carries out the aggregation process. This method promotes expedited convergence of the models for clients equipped with high computational resources. Their proposed solution was evaluated using three clients, each with a pre-trained model for facilitating Transfer Learning. These clients shared an IID version of the CIFAR-10 dataset. In this experimental setup, which resembles centralized learning, their model outperformed current state-of-the-art FL models in terms of convergence speed.

He et al. (2020) [119] introduced the **FedGKt** framework, a system designed specifically for FL involving large CNN models implemented at edge clients. In contrast to traditional methods, **FedGKt** transfers model outputs, termed knowledge, instead of model parameters. The framework operates asynchronously, aggregating information and updating all models using Stochastic GD [120] along with Cross-Entropy [121] and Kullback-Leibler divergence [122] loss functions. Participants in this setup train their models locally and exchange the outputs of the feature extractor and the fully connected layer, as well as the ground truth labels, with the aggregator for each FL CR. The aggregator subsequently updates the global CNN model to accommodate new participants and disseminates the newly predicted labels to all participants. **FedGKt** employs a novel loss function that combines the knowledge of both the aggregator and the participants. **FedGKt** was tested on the CIFAR dataset under both IID and non-IID conditions. The results indicated that despite the asynchronous aggregation process and its inherent flexibility, there was no compromise in terms of model accuracies. Furthermore, the proposed loss function significantly improved learning, offering an average increase of 0.03% compared to both **FedAVG** and centralized learning.

To address the issue of domain shift due to unequal distribution of local datasets, Yao et al. (2022) [55] proposed a federated multi-target domain adaptation (FMTDA) solution known as the DualAdapt framework. In this setup, each participant's device is considered a target domain with an unlabeled dataset, while the aggregator holds a labeled and public dataset that serves as the source domain. The main goal of their approach is to enhance the recognition of outputs from participants' datasets by aggregating their local CNN models. These models share the same feature extraction layers (FELs) but possess different fully connected layers (FCLs) in terms of weights, although all models maintain the same architectural design. During each FL round, the aggregator broadcasts the global model to the participants. Each participant then fine-tunes the FELs, computes their respective FCLs using the maximum classifier discrepancy (MCD) technique [123], encodes its statistical distribution using a Gaussian mixture model (GMM) on its dataset, and shares the FCLs and GMM parameters with the aggregator. The aggregator then updates the FELs and rebroadcasts them throughout the network. Experiments conducted on five-digit datasets [124, 125, 126] and the Cross-City dataset [127] confirmed that **DualAdapt** enhances the average accuracy by 2.4% and requires only a quarter of the computational cost and half

of the communication overhead compared to baseline FL algorithms (**FedAVG**).

Xie et al. (2019) [40] proposed an asynchronous FL aggregation method, known as **FedAsync**, aimed at enhancing the flexibility and scalability of FL and addressing issues related to non-IID data and non-convex problems. After the initialization and broadcasting of the global model, the aggregator awaits any incoming local model and updates the global model using the Gradient Descent (GD) algorithm. To regulate the function, a penalty term is added to the loss function for any local updates. This approach tackles the issue of staleness when participants have heterogeneous computing resources and non-IID data, making FL flexible for any new participant. They evaluated their algorithm on the CIFAR and WikiText-2 [128] datasets using 100 devices, and compared it with **FedAVG**. Their findings showed that **FedAsync** is sensitive to hyperparameters that measure the staleness of participants and significantly depends on them. However, the results indicated that **FedAsync** performs comparably to **FedAVG**, even under high staleness conditions.

Tian et al. (2021) [53] proposed a Delay Compensated Adam (DC-Adam) asynchronous FL approach for anomaly detection in resource-constrained IoT devices using DL techniques. This method consists of three steps. (1) Pre-initialization of global model parameters: A random set of clients transmit a small portion of their data to the server to train the initial global model, which is then distributed to all clients to start FL. This step aims to reduce errors associated with non-IID data. (2) Immediate aggregation and (3) delayed gradient compensation: Upon receiving a local update from a client, the server immediately aggregates the information and compensates for delayed gradients. To evaluate their approach, the authors compared the convergence of loss functions over training epochs on MNIST, CICIDS-2017 [129], and IoT-26 [130] datasets using Sync-Adam, Asynch-SGD, Asynch-Adam, and Asynch-DC-Adam (their proposal) optimizer algorithms. Their results indicated that Sync-Adam and Asynch-DC-Adam outperformed the other optimizers. However, the study did not examine node reliability, and the server faced bandwidth load issues and potential crashes when simultaneously pre-initializing global model parameters, performing aggregation, and compensating for delayed gradients.

Yao and Ansari (2020) [131] proposed a FL enhancement approach for network anomaly detection, based on fog computing. The aim was to accelerate federated learning and minimize energy consumption by controlling the CPU frequency and wireless transmission power (WTP) of all IoT devices. The FL time, comprised of the computation time for local model training and the wireless transmission time for uploading local model updates to the fog node, was designed to meet a Quality of Service (QoS) requirement by not exceeding the maximum permissible FL time. The same concept was applied to energy consumption. To determine the optimal WTP and CPU frequency values for each IoT device, an alternating direction algorithm (ALTD) was implemented. This algorithm iteratively updates the WTP based on previous frequency values and the CPU frequency based on the current WTP value within each local iteration across all IoT devices. The performance of the ALTD algorithm was compared to three other strategies: `power-only`, `CPU-only`, and `fixed`. Across all comparisons, the ALTD method consistently delivered superior results

in terms of energy consumption and learning time, especially as the number of IoT devices, the number of examples in the dataset, the number of CPU cycles, and the size of the dataset increased.

Xu et al. (2019) [132] introduced **ELFISH**, a resource-aware FL framework designed to mitigate the delays experienced by edge devices when sharing their models with an aggregator, primarily due to their physical resource constraints. The main objective was to expedite model training and prevent lagging edge devices from slowing the overall process. To achieve this, they implemented a soft-training method that dynamically masked a calculated number of neurons in each edge device during each training cycle. This procedure included a rule that accounted for the computational workload, training memory usage, training time consumption, and the contribution to the global convergence of each neuron. Furthermore, they introduced a parameter aggregation scheme to recover the masked weights during aggregation, with the goal of enhancing both the accuracy and convergence speed during edge training. The performance of their framework was evaluated by simulating devices with varying computational capabilities and distinct resource constraints, achieved by adjusting CPU bandwidth and memory availability settings. Additionally, pre-trained AlexNet and LeNet models on the MNIST and CIFAR datasets were used in constructing their framework. When compared to conventional synchronized or asynchronous FL methods, ELFISH consistently demonstrated superior accuracy and faster convergence speed.

Saha et al. (2020) [50] proposed a framework, **FogFL**, designed to minimize communication latency, reduce the energy consumption of resource-constrained edge devices, and enhance system reliability. This was accomplished by utilizing a greedy heuristic approach for the selection of an optimal fog unit as a temporary aggregator during FL iteration. The system is arranged as a set of edge device clusters, each linked to the nearest fog unit. After a number of local training iterations, which are adjusted based on the physical characteristics of the edge devices, the weight vectors are forwarded to the fog aggregator. Subsequently, each fog unit sends workload and communication latency parameters to the cloud server, which selects the aggregator based on these parameters. To evaluate their proposal, they deployed participant nodes with varying physical attributes and technologies, and the non-IID MNIST dataset. They compared the **FogFL** results with the **FedAVG** algorithm and the hierarchical FL framework (**HeFL**) [133]. The results indicated that **FogFL** reduced delay by 85% and 68% compared to the **FedAVG** and **HeFL**, respectively, decreased energy consumption by 92% compared to **FedAVG**, and reduced the number of CRs necessary to complete the target global model accuracy.

Khan et al. (2020) [134] proposed an FL approach that is specifically designed for edge networks. Their approach utilizes the Stackelberg game [135] to encourage device participation in the FL process, by offering rewards while taking into account both communication and computation costs for both the edge server and edge devices. Their system comprises an edge server and a set of user devices with non-IID data, each with different computation and communication resources. Their implementation of the Stackelberg game enables

the selection of a subset of IoT devices eligible for participation in the FL process with an aim to minimize overall training costs. It also provides for differential rewards to be offered based on the distinct training costs of the devices. To assess their incentive-based FL model, they used multinomial logistic regression and divided the MNIST dataset into five user groups, each characterized by unique communication channel properties. They found that higher reward rates led users to perform more iterations within a single global iteration, thereby enhancing accuracy. However, their proposal does not elaborate on how reward allocation and optimal use of training resources are determined.

Qu et al. 2020 [31] introduced **D2C** for big data-cognitive computing in Industry 4.0 networks. **D2C** incorporates a decentralized paradigm based on blockchain-enabled FL to enhance the performance of Industry 4.0 manufacturing systems. The approach looks to guarantee data security and efficient processing, and provides incentive mechanisms to encourage participation in the learning process, while also helping to mitigate poisoning attacks. In each FL round, model weights are sent to a cluster of miners, who verify the authenticity of local model parameters via a cross-verification mechanism. The Proof of Work (PoW) consensus algorithm assigns a target nonce for each round. The miners continuously generate random nonces until the target is found. Once identified, the process is halted, and the successful miner is rewarded, thus promoting further participation in the learning process. This miner obtains the right to use the associated block as the new block and subsequently broadcasts it to all other parties. Local model parameters are then aggregated using the Distributed Approximate Newton (DANE) method [136]. The newly computed global parameters are stored in a block and made accessible for all machines to download. The proposed framework was tested on CNN models trained on the CIFAR-10 dataset under IID conditions. The results showed that the D2C framework boosts global accuracy by 0.12% compared to the baseline FL model and facilitates faster convergence across various learning rates.

Liu et al. (2021) [137] introduced a distributed FL and blockchain-based framework named Vehicle Intrusion Detection System (**FL-VIDS**). It emphasizes the preservation of vehicle privacy using differential privacy and aims to curtail communication overhead and computational expenses by obviating the need for a global server, while establishing a secure model-sharing protocol among edge devices, particularly the Roadside Units (RSUs). RSUs in this setup, gather data from vehicles traversing the same vicinity and act as local aggregators. They undertake the execution of FL and securely store the consequent models in the blockchain. A unique facet of this system is the competitive dynamic fostered for the authentication of aggregated model training transactions. The RSU winning this competition incorporates these transactions into the blockchain via a distributed consensus process and evaluates trustworthiness based on the accuracy of the model. To counteract potential malevolent attacks, homomorphic encryption is deployed. The system's efficacy was assessed using the DDCup99 dataset [138] with a multi-layer perceptron model. The results revealed a direct correlation between the dataset's size and model accuracy, and temporal complexity, while showing an increment in model accuracy with an increased

number of collaborating nodes. Significantly, the proposed Proof of Authority (POA) mechanism was found to bolster trust and diminish mining complexity.

6.1. Discussion and comparison

In this section, we studied various FL aggregation methodologies developed between 2016 and 2022. This exhaustive examination of recent contributions illuminates how they tackle prevalent FL challenges such as statistical heterogeneity, system heterogeneity, expensive communication, and privacy concerns. The surveyed contributions related to FL aggregation methods are systematically organized and contrasted in Table 6.1. Differentiating criteria include the challenges addressed: statistical heterogeneity (c1), system heterogeneity (c2), expensive communication (c3), and privacy concerns (c4). Additional features considered are the deployment setting of FL (Cloud, Fog, or Edge), the learning model, the dataset employed for simulation, foundational solution strategies, and techniques to ensure privacy.

From the evidence provided in Table 6.1, it becomes apparent that confronting statistical heterogeneity is a focal point in FL. The prime objective of FL is to refine decision-making processes by training a global AI model that leverages the diversity and distribution of data across multiple participants, all while ensuring data privacy. In practical scenarios, data is often non-IID and originates from disparate domains, potentially leading to a deceleration in learning convergence. Furthermore, numerous FL contributions adhere to the deployment of a single AI model architecture across participant devices. This strategy could oblige participants with limited resources to contribute, potentially prolonging system processing times and instigating issues related to computational and communication resources. Consequently, FL contributions primarily center on mitigating this challenge, thereby enhancing the performance and effectiveness of both global and local AI models within the FL system.

Several strategies have been proposed to address the challenge of handling non-IID data in FL. One such strategy includes executing FL aggregation on the base layers of the NN model, rather than on the entire model [48, 30]. However, this approach can only be applied when dealing with homogeneous local models. Another strategy utilizes clustering techniques to group similar participants based on the characteristics of their data, thereby improving the management of the distributed learning process and enhancing model performance [49, 50, 51, 114]. Normalization of models has also been employed to ensure a consistent representation across the models [52]. Alternatively, using a public dataset, consisting of subsets of the local datasets, allows for a more realistic depiction of the data distribution, especially when the data falls within the same domain [53, 139]. Integration of meta-learning [110] into FL systems can also boost their flexibility. This enables supporting new participants, even when they have limited and non-IID data [109]. Finally, to address the challenge of domain shift [79] and the evolving nature of unsupervised local datasets within the FL context, domain adaptation techniques [140] have been employed. These techniques have proven effective in enhancing the performance and generalization

capabilities of models across diverse data distributions from multiple participants in the FL system [55, 56].

Knowledge distillation is an alternative to address the heterogeneity of local models in Fl. By using a public dataset as a benchmark in FL and sharing the predictions of AI models instead of their parameters, this approach can effectively mitigate the issues associated with local model heterogeneity while also ensuring data privacy [58, 59, 57, 134]. However, these solutions might not always be applicable in real-world scenarios due to the presence of multi-domain data [139] or multitask learning [141] among participants, which could introduce inconsistencies in the benchmark dataset within FL strategies. To address this challenge, further research and development is required to devise methods capable of handling diverse data distributions in practical applications.

Addressing the challenge of system heterogeneity in FL can be accomplished by employing robust aggregation methods. These methods encompass asynchronous algorithms [40, 53, 134], incentive mechanisms [134], strategies employing dynamic and multiple aggregators [50], as well as approaches adapting local training objectives [4, 54]. However, asynchronous algorithms accommodate disparities in computational capabilities and network conditions by processing and aggregating device updates at disparate time intervals [40, 53, 134]. These algorithms can potentially induce convergence issues due to inconsistent updates to the global model by participating devices. Moreover, they may complicate the real-time monitoring and assessment of the overall FL system performance. Thus, incentive mechanisms [134] encourage balanced participation by offering rewards. Nevertheless, they may introduce overheads, such as increased communication costs for reporting rewards and monitoring contributions. Further, these mechanisms are susceptible to exploitation by malicious or selfish participants who claim rewards without making meaningful contributions. Furthermore, the use of dynamic and multiple aggregators [50] distributes the aggregation workload, thus mitigating the impact of heterogeneity on the overall model performance and managing bandwidth overhead. Yet, this strategy may necessitate more computational resources and potentially result in increased communication overhead and potential bandwidth congestion, particularly when network resources are limited. In addition, adaptive local training objectives tailor learning goals according to the unique properties of each device's local dataset [4, 54]. This allows FL systems to better handle diverse data distributions and produce a more accurate global model. However, they may slow down the learning process due to proximal term calculations and hyperparameter determination.

The issue of high communication costs is a prevalent challenge, while current aggregation methods may not tackle this directly, they incorporate a variety of techniques into their design to offer broader solutions. These solutions are intended to minimize communication costs while enhancing the overall performance of the FL. Additionally, privacy concerns represent a significant hurdle in this area, necessitating the involvement of sophisticated cybersecurity techniques adopted for FL. This has been predominantly addressed in recent times through the adoption of blockchain techniques [137, 31]. These techniques strive to

ensure robust data privacy and security during the exchange of local and global DL model parameters, mitigating potential exacerbations of existing FL challenges. Notably, the exchange of DL models introduces more complexity than conventional message exchanges via blockchains, which could potentially impair the efficiency and efficacy of FL systems.

Furthermore, most FL contributions are implemented on cloud infrastructures, taking full advantage of the massive computing resources that they provide. However, there is a distinct shift towards edge- and fog-based deployments. The growing popularity of edge [134, 31, 137] and fog-based [131, 50] deployments can largely be attributed to major features such as the proximity of model parameters, which facilitates quick, efficient parameters access, a significant decrease in latency, which improves real-time interaction, and more control over privacy to provide optimal data and parameters security.

In conclusion, it is evident that most of the developed aggregation algorithms in FL are designed with a focus on generalizability. This allows their application across a broad spectrum of real-world scenarios, taking into account their respective constraints and advantages. These algorithms have undergone rigorous testing and validation on various benchmark datasets, such as the LEAF datasets [142], and with pre-existing models such as CNN and LSTM. This process ensures their effectiveness and adaptability across different contexts. Such generalizability empowers researchers and professionals to confidently adopt these algorithms and customize them to meet the specific needs of their FL applications.

!

Reference	Year	Contribution	Challenge				Deployment	Learning Model	Dataset	Solution-based	Privacy
			c1	c2	c3	c4					
[107]	2016	FedSGD	×	×	×	×	Cloud-based	CNN & LSTM	MNIST [124] & SHAKESPEARE[143]	Gradients Averaging	×
[4]	2017	FedAVG	×	×	√	×	Cloud-based	CNN & LSTM	MNIST [124] & SHAKESPEARE[143]	Weights Averaging	×
[109]	2018	FedMeta	√	×	√	√	Cloud-based	CNN & ANN	LEAF [142] & real-world production	Meta-Learning	HE
[48]	2019	FedPer	√	×	√	×	Cloud$_b$*ased*	ResNet-34 & MobileNet-v1	CIFAR [144] & FLICKR-AES [145]	Personalized FL averaging	×
[57]	2019	FedMD	√	×	×	×	Cloud$_b$*ased*	CNN	LEAF [142] & CIFAR [144]	TL and knowledge distillation	×
[40]	2019	FedAsyn	√	√	×	×	Cloud$_b$*ased*	CNN & LSTM	CIFAR [144] & WikiText-2 [146]	Asynchrounous FL	×
[54]	2020	FedProx	√	√	×	×	Cloud$_b$*ased*	MLR & LSTM	LEAF [142] & SHAKESPEARE[143]	Models divergence	×
[111]	2020	FedMA	√	×	×	×	Cloud$_b$*ased*	CNN & LSTM	LEAF [142] & SHAKESPEARE[143]	Layer-wise Averaging	×
[134]	2020	FedGKT	√	√	×	×	Cloud$_b$*ased*	CNN	CIFAR [144]	Asynchrounous Knowledge Transfer	×
[49]	2020	FL+HC	√	×	×	×	Cloud$_b$*ased*	CNN	FEMNIST [126]	Agglomerative Hierarchical Clustering	×
[131]	2020	ALTD	×	√	×	×	Fog-based	ANN	Generated	Computational Ressources Optimization	×
[132]	2019	ELFISH	×	√	√	×	Cloud$_b$*ased*	Alexnet & LeNet	MNIST [124] & CIFAR [144]	Models compression & soft-training	×
[50]	2020	FogFL	√	√	√	×	Fog-based	CNN	MNIST [124]	Greedy Heuristic approach	×
[134]	2020	Stackelberg+FedAVG	√	√	√	×	Edge-based	CNN	MNIST [124]	Stackelberg Game Mechanism	×
[31]	2020	D2C	√	×	×	√	Edge-based	CNN	CIFAR [144]	DANE [136]	Blockchain
[137]	2021	FL-VIDS	×	×	×	√	Edge-based	MLP	DDCup99 [138]	Models Averaging	DP & Blockchain
[51]	2021	FedDist	√	×	×	×	Cloud-based	CNN	LEAF [142]	Distance Similarity & Layer-wise Traning	×
[53]	2021	Asynch-DC-Adam	√	√	×	×	Cloud-based	CNN & ANN	MNIST [124] & CICIDS-2017 [129] & IoT-26 [130]	ADAM Optimization in Asynchronous FL	×
[59]	2021	MHAT	√	×	√	×	Cloud-based	CNN	MNIST [124]	TL and knowledge distillation	×
[58]	2021	FedH2L	√	×	√	×	Cloud-based	CNN	MNIST [124] & Office Home [147]	TL and knowledge distillation	×
[118]	2021	FTL+FedAVG	√	√	×	×	Cloud-based	CNN	CIFAR [144]	Hierarchical FL	×
[30]	2022	FedGA	√	×	√	×	Cloud-based	CNN	MNIST [124]	Genetic Algorithm	×
[52]	2022	FTL-RLS	√	√	√	×	Cloud-based	ANN	Generated	Recursive Least Squares (RLS) algorithm	×
[113]	2022	FedFM	√	×	√	×	Cloud-based	RestNet	CIFAR [144]	Anchor-based Feature Matching	×
[113]	2022	FedSim	√	×	√	×	Cloud-based	CNN	FedME-x [113] & Fed-Goodread [113] & EMNIST [126] & MNIST [124]	Principal Component Analysis	×
[55]	2022	FMTDA	√	√	√	×	Cloud-based	CNN	five digits & Cross-City [127]	Domain Adaptation	×

Table 6: Summary of related works on FL aggregation methods, where c1, c2, c3, c4 means statistical heterogeneity, system heterogeneity, expensive communication, and privacy concerns, federated learning challenges.

7. Federated Learning Development Tools

Before starting the process of designing a FL framework, it is necessary to first specify a set of development tools that are appropriate for the desired solution. This needs to be done in accordance with the features of the problem, the specifications of the environment, and the limitations of the solution. Either build a FL tool from scratch, which requires fundamental and unadulterated tools like PyTorch or TensorFlow, or reuse existing frameworks in order to use the algorithms they provide and modify them if they are open source. In addition, we need, among the many tools for the creation of FL, a dataset or a set of datasets that matches to the specifications of participants' systems. The following will serve to describe all of these specifics in more detail.

7.1. FL Datasets

FL is developed to locally store private data, as previously indicated. So, most FL researchers evaluate and validate their algorithms using benchmark datasets. Even if we find a real FL research use case, their datasets are not sharable since they concern the confidentiality of their participants (e.g. patients, employees). That's why, many researchers and industrials create benchmark datasets to get creative with FL. According to state-of-the-art studies, we found some repetitive datasets, which are represented in Table 7 and they are detailed as follows, and compared in Table 8.

7.1.1. MNIST (Modified National Institute of Standards and Technology dataset)

MNIST [124] dataset is commonly used to benchmark image recognition and ML algorithms. It has several size-normalized and centered handwritten digits. Each MNIST image is represented by 28x28 pixels and contains a grayscale digit from 0 to 9. The dataset includes 60,000 and 10,000 training and testing images respectively. Also, there are some related datasets such as:

- USPS [125], which contains 7438 training and 1860 testing respectively 16 $\times$16 gray and blurry digit images.
- The Street View House Numbers (SVHN) [148] dataset is a substantial collection of real-world images sourced from house numbers visible in Google Street View images. This dataset was compiled by researchers at Stanford University in collaboration with Google. It comprises over 73,257 digits for training, 26,032 digits for testing, and an additional 531,131 samples that are somewhat less challenging, and intended to be used as extra training data. Each image is a 32x32 RGB image, clearly annotated with a bounding box and the number it represents.
- The MNIST-M [140] dataset is composed of 60,000 images used for training and 10,000 images used for testing. Each 28x28 pixels image is extracted from MNIST dataset with color transformation (RGB) by adding background noise to make it complex.
- The Extended Modified National Institute of Standards and Technology(EMNIST) [140] dataset is bigger and has more variety than the original MNIST dataset. Each image in the EMNIST dataset is 28x28 pixels. It has more than 800,000 handmade letters, numbers, and punctuation marks, including both uppercase and lowercase letters. This makes it more complicated and different from the original MNIST collection.

7.1.2. Fashion MNIST

FashionMNIST [149] replaces MNIST by containing images of clothing and accessories instead of handwritten numbers. It contains 28x28 grayscale images of one fashion item, such as a shirt, purse, or shoe. The dataset contains 60,000 and 10,000 training and testing images respectively. It is more complex than MNIST because it contains different patterns and forms for sample representation.

7.1.3. Leaf

Leaf [142] is a benchmark that includes a collection of open-source federated datasets with different types of data. Each of them groups a set of samples collected from different users. It is used to evaluate FL, multitask learning, and meta-learning frameworks and algorithms. Its sub-datasets are represented as follows.

- Federated Extended MNIST (FEMNIST) is a derivative of the EMNIST dataset. It includes grayscale images of 28x28 pixels, each representing handwritten digits, as well as both upper and lowercase characters. What sets FEMNIST apart is that these images are distributed across multiple clients, thus simulating a realistic FL scenario.
- Sentiment140 dataset is a widely-used dataset for sentiment analysis and opinion mining [150]. It consists of 1.6 million tweets annotated with emoticons indicating positive, negative, and neutral sentiments. The tweets in the Sentiment140 dataset are written in English and cover a wide range of topics, including politics, sports, entertainment, and technology.
- Shakespeare dataset refers to a collection of texts written by the English playwright and poet William Shakespeare [143]. The dataset typically includes all of Shakespeare's known works, including his plays, sonnets, and poems. The dataset is used to train neural network models that predict the next character of a word.
- CelebA is a large-scale face attributes dataset consisting of over 202,599 celebrity images [151], each with 40 attribute annotations. The dataset was produced specifically for the purpose of using it in facial attribute recognition activities, such as face categorization, attribute prediction, and face modification.
- Reddit dataset is a collection of data from the popular social media platform [152]. This data typically includes information such as user comments, submissions, and upvote/downvote counts. It can be obtained through the Reddit API or by web scraping, although access to the full dataset can be limited due to size constraints and Reddit's terms of service.

7.1.4. CIFAR-100 & CIFAR-10

The CIFAR-100 [144] dataset was developed by the Canadian Institute For Advanced Research. It is widely used as a benchmark for image classification and computer vision applications. it is a labeled subset of the 80 million small images dataset. They're all labeled with one of 100 fine-grained classes that are combined into 20 coarse-grained classes. Also, CIFAR-10 dataset extends CIFAR-100 dataset. It consists of 60,000 32x32 color training images and 10,000 test images, with 10 classes.

7.1.5. Quick, draw!

The Quick, Draw! dataset is a collection of hand-drawn sketches created by users of the Quick, Draw! game, an online game [153] developed by Google that challenges players to draw a picture of an object in 20 seconds. The game records the sketches and stores them in the Quick, Draw! dataset, which contains over 50 million drawings across 345 categories, such as animals, objects, and symbols.

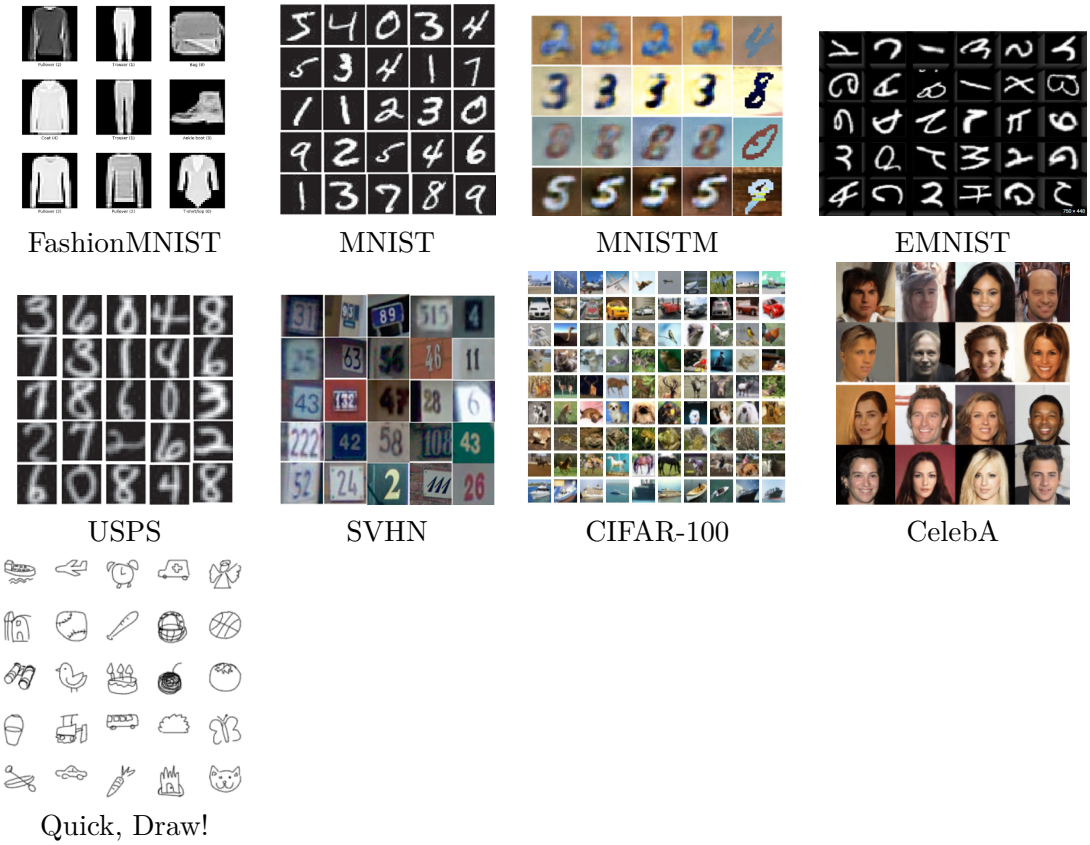

Table 7: Examples of FL benchmark computer vision datasets.

Comparison

The features of numerous benchmark datasets frequently used for FL simulations are listed in 8. These datasets cover a wide range of topics, including sentiment analysis in text, handwritten digits, home numbers, apparel, and more. There are differences between them in terms of the number of classes, the types of data they contain (text, images in grayscale or color), the size of the sample set, the dimensions of the data, and the amount of preprocessing necessary. Several datasets, especially MNIST, USPS, MNIST-M, and EMNIST, are centered on handwritten digits, with the latter extending to handwritten letters. They are all grayscale pictures that require little preprocessing. The SVHN dataset, on the other hand, contains color images and requires just a little preparation. The Sentiment140 dataset, collected from Twitter, requires considerable preprocessing in the text domain, including tasks like tokenization and stop word removal. Similarly,

the Reddit dataset, which contains billions of comments, necessitates considerable preparation in order to anonymize the data and remove stop words.

In summary, This wide array of datasets meets the diverse needs of FL simulations and depends on the research problem and, consequently, the type of data that needs to be processed. Our FL review revealed that a large portion of the contributions was evaluated using images because of their complex pre-process and NN models. In such cases, the proof of algorithms on increasingly complicated scenarios would imply that they are adaptable to every scenario.

Dataset	Domain	Classes	Data Type	Sample size	Dimensions	Preprocessing Required
MNIST [124]	Handwritten Digits	10	Grayscale Images	60,000 Training, 10,000 Testing	28x28	Minimal
USPS [125]	Handwritten Digits	10	Grayscale Images	7,438 Training, 1,860 Testing	16x16	Minimal
SVHN [148]	House Numbers	10	Color Images	73,257 Digits for training, 26,032 Digits for testing	32x32	Moderate (e.g., bounding box localization)
MNIST-M [140]	Handwritten Digits	10	RGB Images	60,000 Training, 10,000 Testing	28x28	Minimal
EMNIST [140]	Handwritten Digits Letters	62 (10 digits + 26*2 letters)	Grayscale Images	800,000	28x28	Minimal
Fashion MNIST [149]	Clothing	10	Grayscale Images	60,000 Training, 10,000 Testing	28x28	Minimal
FEMNIST [142]	Handwriting	62 (10 digits + 26*2 letters)	Grayscale Images	3,550,000	28x28	Minimal
Sentiment140 [150]	Text Sentiment	2 (Positive, Negative)	Text	1,600,000 Tweets	Variable Length	Yes (e.g., tokenization, stop word removal)
Shakespeare [143]	Text	N/A	Text	35 Plays, 154 Sonnets	Variable Length	Yes (e.g., tokenization, parsing)
CelebA [151]	Faces	40 Attributes	Color Images	202,599	218x178	Moderate (e.g., cropping, scaling, face alignment)
Reddit [152]	Textual Conversations	N/A	Text	Billions of comments	Variable Length	Yes (e.g., tokenization, stop word removal, anonymization)
CIFAR-10 [144]	Objects	10	Color Images	50,000 Training, 10,000 Testing	32x32	Minimal
CIFAR-100 [144]	Objects	100	Color Images	80 million Images	32x32	Minimal
Quick Draw [153]	Sketches	345	Strokes as coordinates	50 million	Variable Length	Yes (e.g., normalization, scaling)

Table 8: Comparison of Benchmark Datasets used for Federated Learning Simulation.

7.2. FL Frameworks

The realm of FL has witnessed the development of various dedicated tools and frameworks, each incorporating a range of FL algorithms. These tools are strategically designed to facilitate the seamless implementation of FL in practical, real-world scenarios. Consequently, the objective of this section is to introduce and discuss a selection of these practical frameworks that significantly contribute to the ongoing advancement and application of FL.

7.2.1. Federated AI Technology Enabler (FATE)

FATE [154] is the first industry-grade open-source framework for FL, launched in February 2019. It enables companies and institutions to collaborate on data while protecting data security and privacy. It supports HFL and VFL categorization and implements secure computing protocols based on DP and SMPC. Also, it supports a number of FL algorithms, including logistic regression, tree-based algorithms, DL, and TL.

7.2.2. Substra

Substra [155] is a Owkin project centered on data ownership and privacy, providing an enterprise solution tailored for healthcare. With various interfaces, including a Python library, command-line interfaces, and a graphical UI, it caters to a wide range of users. The framework's commitment to privacy, traceability, and security is reinforced through encryption for model updates, data storage, and network communication. This makes Substra an apt choice for sensitive sectors like healthcare. It accommodates DL models, employs HFL for data partitioning, and uses SMPC as a security measure.

7.2.3. PyTorch

PyTorch [156] is a widely-used, open-source ML framework developed by Facebook's artificial intelligence research group. Based on Python, it provides comprehensive support for the construction and training of neural networks. Notably, PyTorch is flexible and capable of supporting large-scale deployments on mobile devices, and all data partition types. It facilitates FL by offering resources and tools to implement this technique from scratch, such as benchmark datasets and security libraries. Moreover, PyTorch extends its commitment to security by incorporating SMPC and DP as security strategies.

7.2.4. PySyft

PySyft [157] is a privacy-focused Python-based DL library. Being built on top of the PyTorch framework, it inherits some of PyTorch's FL functionalities, such as data partitioning. PySyft supports secure computations through encrypted computation and DP mechanisms to safeguard data, and it employs both dynamic and static graphs of computations. Additionally, PySyft provides HE as a security approach. Despite these capabilities, PySyft is primarily used for simulations and as such, it may not offer the flexibility and support for large-scale collaborators found in some other frameworks.

7.2.5. PyGrid

PyGrid [158] serves as a management and deployment API for PySyft, tackling real-world data science challenges. It facilitates FL across various platforms, including the web, mobile devices, and edge devices, providing the infrastructure to distribute and orchestrate computations across multiple nodes or machines.

7.2.6. Flower

Flower is an open-source and adaptable FL framework developed by the Systems and Networking Group at the University of Oxford [159]. Compatible with various ML libraries including TensorFlow, PyTorch, and Keras, it supports an array of real-world FL scenarios, demonstrating flexibility and extending support to mobile devices. As for secure aggregation, Flower incorporates the SecAgg+ protocol [160], although it primarily supports HFL.

7.2.7. Open Federated Learning

OpenFL [161], developed by Intel, is an open-source Python compatible framework that supports scalable FL. It primarily caters to HFL. The framework comprises two main components: the Collaborator, representing the end device, and the Aggregator. OpenFL provides both a Python API and a Command-Line Interface, accommodating diverse modes of user interaction. Security for inter-node communication is ensured through mutual TLS (mTLS), a protocol enabling bidirectional authentication among all nodes within the federation, each of which must be certified. OpenFL also has data compression capabilities, offering both lossy and lossless options to optimize data transmission. Furthermore, it operates within Docker containers, isolating federation environments for enhanced security and reproducibility of FL tasks.

7.2.8. TensorFlow Federated (TFF)

TFF [4] is a Python open-source framework developed by Google that implements FL to trait decentralized data on multiples servers; it shares a global model to clients. Also, it enables developers to use the included FL algorithms. There are two types of TFF layers: federated learning API, which is a layer that allows developers to use the existing TensorFlow models in order to implement FL. It consists of three main parts: `Models`, which are classes and helper functions that enable the wrapping of existing models with TFF, `Federated Computation Builders` that are the helper functions to construct federated computations, and `Datasets`, which are used for simulation scenarios. Federated Code API, which is a layer that allows developers to express novel federated algorithms by combining TensorFlow with distributed communication operators.

7.2.9. IBM FL

IBM FL [162] is a versatile Python framework for FL, developed by IBM. It provides the necessary resources for collective model training, independent of any single ML framework, thereby supporting a variety of learning topologies. Both DL and traditional ML techniques are accommodated, spanning supervised, unsupervised, and RL methodologies. With HFL as a data partitioning scheme and SMPC and DP as security approaches, IBM FL is flexible and robust. It extends support to mobile devices and major companies, making it viable for real-world scenarios.

7.2.10. NVIDIA Clara

NVIDIA Clara [163] is a platform designed for the innovation and deployment of AI-powered imaging, genomics, and smart sensor solutions. It enables developers, data scientists, and researchers to build real-time, secure, and scalable solutions using comprehensive GPU-accelerated frameworks, SDKs, and reference applications. NVIDIA Clara leverages AutoML, privacy-preserving FL, and TL for the construction of sophisticated DL models. It supports HFL as a data partitioning strategy, and utilizes SMPC and DP for ensuring secure computations. Demonstrating flexibility, NVIDIA Clara primarily extends its support to powerful hospitals, highlighting its real-world application in global healthcare scenarios.

7.2.11. Federated Machine Learning (FedML)

FedML [164] is an open-source research framework designed to enable researchers and developers to experiment with various FL algorithms and benchmark their performance against others. It offers a range of experimental datasets and models, along with tools for performance measurement and result analysis. FedML supports HFL, VFL, and FTL as data partitioning strategies. Additionally, it employs DP and SMPC as security techniques. The framework is inclusive of numerous state-of-the-art aggregation algorithms, and caters to three computing paradigms: distributed computing, standalone simulation, and mobile on-device training.

Comparison

Table 9 presents a detailed comparison of various FL frameworks. Each framework is evaluated on several key parameters, including the maintainer, language, support for different ML models, support for different FL data partition types (HFL, VFL, FTL), scalability, mobile support, privacy considerations, and the availability of documentation. These frameworks are maintained by diverse companies, each contributing distinct technologies and solutions for FL deployment. Some frameworks like IBM FL, OpenFL, and FedML offer a high level of abstraction, providing predefined solutions where developers simply choose the ones suitable for their application. Although this can streamline the development process, it can also limit developers' flexibility and enforce certain constraints. Alternatively, frameworks such as PyTorch and TensorFlow, when used in combination with libraries like PySyft and TFF, offer the flexibility of building FL solutions from scratch. These platforms provide support for all FL types and offer a variety of methods and approaches for implementing FL. Regarding privacy, each of the compared frameworks implements robust techniques to ensure data protection. For instance, PyTorch and FedML use SMPC and DP, while Flower leverages SecAgg+ and OpenFL uses mutual TLS for secure internode communication. When it comes to documentation, all frameworks offer comprehensive guides, enabling developers to understand the framework's architecture and operation. This reflects that these frameworks are well-supported and actively maintained by their respective communities or companies

7.3. Federated Learning Evaluation Metrics

Metrics for evaluating FL solutions can be put into a number of categories, such as communication efficiency, contribution measurement, model performance, and assessment of the whole training process.

7.3.1. Communication efficiency

Evaluating the communication efficiency [165] of the FL algorithms is one of the key objectives in this area of study. During the training process, this may be evaluated by measuring the amount of data that is communicated between participant devices and the central server and the number of FL CR needed to achieve the target goal. This has the potential to have a major impact on the performance of the system as a whole.

7.3.2. Contribution measurement

When participating in FL, it is absolutely necessary to evaluate the contribution made by each participant to the overall process [166]. This assists in determining credits fairly and ensures that each party's contributions are correctly appreciated. One way that contribution can be measured

is by an analysis of the effect that the data contributed by each participant has on the accuracy of the global model as a whole.

7.3.3. Model performance

Evaluating the model performance is done by determining the accuracy, convergence rate, generalization ability, and effectiveness of the global model that was developed through the FL process and training using decentralized IID or non-IID data from many participants [4]. Also, Benchmarking different FL algorithms can help assess its performance, such as FedAVG [4] and centralized AI approach.

7.3.4. Computation overhead

The computation overhead metric determines how much computation time, and resource utilization are required for each participant device, as well as the servers that have to be used in order to complete the tasks that came with the FL system [167]. This metric can be measured by MIPS (Million Instructions Per Second) and FLOPS (Floating Point Operations Per Second) [19], which are general performance parameters for computers and processors [168].

7.3.5. Privacy and Security robustness

When evaluating the robustness of privacy and security in FL, it is necessary to first get a comprehension of the potential vulnerabilities and threats, and then evaluate the efficacy of the solutions that have been implemented to tackle these risks [169]. This can be measured by evaluating the resilience of the developed FL system against various attacks that are mentioned in Section 5.3.

7.4. Discussion

In this section, we have mentioned the final steps in the process of developing a FL technique, including the use of benchmark datasets like MNIST [124] in order to validate the developed algorithms, knowing that the data is private in the context of FL and therefore the validation is done on benchmarks and the application is done on real use cases. Then, the technical development of FL is implemented using frameworks and libraries such as TFF [4] which are used to apply existing techniques or the development from scratch of new tools. The choice of such a framework depends on the needs and requirements of the developed FL method. Finally, the most critical step is the evaluation of such a technique using a set of measures such as the evaluation of the communication and computational overload, the level of contribution of the participants, the performance of the model, and the robustness of the system against privacy and security challenges.

Framework	Reference	Maintainer	Language	ML model	HFL	VFL	FTL	Scalability	Mobile support	Privacy concerns	Documentation
Fate	[154]	Webank & Linux Foundation	Python	√	√	√	×	√	×	SMPC, DP	√
Substra	[155]	Owkin	Python	DL	√	×	×	√	×	SMPC	√
PyTorch	[156]	Facebook	Python	√	√	√	√	√	√	SMPC, DP	√
Pysyft	[157]	OpenMined	Python	DL	√	√	√	×	×	SMPC, HE	√
PyGrid	[158]	OpenMined	Python	DL	√	√	√	√	×	SMPC, HE	√
Flower	[159]	Adaptscale	Python	√	√	×	×	√	√	SecAgg+	√
OpenFL	[161]	Intel	Python	√	√	×	×	√	×	Mutual TLS	√
TensorFlow Federated (TFF)	[4]	Google	Python	Tensorflow ML models	√	√	√	√	√	DP	√
IBM FL	[162]	IBM	Python	√	√	×	×	√	√	DP, SMPC	√
NVIDIA Clara	[163]	NVIDIA	Python, C++	Medical Imaging DL models	√	×	×	√	×	SMPC, DP	√
FedML	[164]	FedML Community	Python	√	√	√	√	√	√	SMPC, DP	√

Table 9: Comparison of Federated Learning Frameworks

8. Discussion and perspectives

This paper has provided a comprehensive and systematic review of the FL, covering a wide range of topics including the FL categories, challenges, aggregation methods, and development tools. In particular, we have discussed the challenges of FL, including communication costs, system heterogeneity, statistical heterogeneity and privacy concerns. We have also evaluated and compared the effectiveness of the existing aggregation algorithms such as FedAVG, FedProx, and FedMA in addressing these challenges.

Moreover, this review paper has included an extensive discussion of the benchmark datasets, frameworks and evaluation metrics used to assess the performance of FL systems. By providing an overview of these tools, we aim to provide a valuable resource for new researchers and developers in the field of FL, enabling them to make informed decisions about the tools and techniques that best suit their specific needs.

- **Future Research Directions:** As FL continues to evolve, it is crucial to explore new aggregation algorithms and techniques that can address the diverse challenges faced by this field. These may include novel approaches to handle increasingly complex and heterogeneous data, as well as innovative methods to deal with security and privacy concerns. Researchers should also focus on developing more robust and scalable solutions that can work efficiently across different domains and applications.

- **Cross-Domain Adaptation:** One possible avenue for future research is to investigate the potential of applying FL to cross-domain adaptation scenarios. This would involve exploring how FL techniques can be used to transfer knowledge across various domains and adapt models to new, previously unseen data distributions.

- **Integration of Advanced Techniques:** Integrating advanced ML techniques, such as meta-learning, reinforcement learning, and transfer learning, with the FL can potentially lead to significant improvements in model performance, generalization, and adaptability. Investigating these integrations can pave the way for more powerful and efficient FL systems.

- **Real-world Applications:** With the growth of the FL, it is essential to consider its applicability in real-world scenarios, such as healthcare, finance, smart cities, and industrial IoT. Developing and validating the FL solutions for these domains can help address critical challenges and demonstrate the practical utility of this technology.

- **Standardization and Benchmarking:** As the FL research advances, it is crucial to establish standard benchmarks and evaluation metrics that allow for the fair and consistent comparison of different algorithms and approaches. This will facilitate the identification of best practices and promote the development of more effective FL systems.

In summary, this paper has delivered a comprehensive understanding of the current state of FL by synthesizing key findings from various survey papers. By identifying the strengths, limitations, and challenges in the existing literature, this work serves as a robust foundation for future research and development in this rapidly evolving field. Additionally, by highlighting potential future directions and opportunities, it encourages researchers to explore innovative solutions and applications, further contributing to the growth and maturity of FL as a transformative technology.

Appendix A. List Of abbreviation

IoT Internet of Things

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

FL Federated Learning

TL Transfer Learning

NN Neural Network

RL Reinforcement Learning

MLP Multi-Layer Perceptron

MLR Multinomial Logistic Regression

CNN Convolutional Neural Networks

LSTM Long short-term memory

GD Gradient Descent

IID Independent and Identically distributed

HFL Horizontal Federated Learning

VFL Vertical Federated Learning

FTL Federated Transfer Learning

P2P Peer to Peer

CR Communication rounds

ID Identity

HE Homomorphic Encryption

DP Diffirential Privacy

SMPC Secure Multi-Party Computation

SLR Systematic Literature Review

RQs Research Questions

API Application Programming Interface

References

[1] Mustafa Alper Akkaş and Radosveta Sokullu. An iot-based greenhouse monitoring system with micaz motes. *Procedia computer science*, 113:603–608, 2017.

[2] Iqbal H Sarker. Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN Computer Science*, 3(2):1–20, 2022.

[3] Marija Jegorova, Chaitanya Kaul, Charlie Mayor, Alison Q O'Neil, Alexander Weir, Roderick Murray-Smith, and Sotirios A Tsaftaris. Survey: Leakage and privacy at inference time. *arXiv preprint arXiv:2107.01614*, 2021.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[5] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[6] Fanglan Zheng, Kun Li, Jiang Tian, Xiaojia Xiang, et al. A vertical federated learning method for interpretable scorecard and its application in credit scoring. *arXiv preprint arXiv:2009.06218*, 2020.

[7] Apple Machine Learning Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 1(8), 2017. URL https://machinelearning.apple.com/research/learning-with-privacy-at-scale.

[8] Angela Carrera-Rivera, William Ochoa-Agurto, Felix Larrinaga, and Ganix Lasa. How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX*, page 101895, 2022.

[9] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

[10] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7(2):639–657, 2021.

[11] Momina Shaheen, Muhammad Shoaib Farooq, Tariq Umer, and Byung-Seo Kim. Applications of federated learning; taxonomy, challenges, and research trends. *Electronics*, 11(4): 670, 2022.

[12] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges. *arXiv preprint arXiv:2211.08413*, 2022.

[13] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, pages 1–23, 2022.

[14] Sharnil Pandya, Gautam Srivastava, Rutvij Jhaveri, M Rajasekhara Babu, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Spyridon Mastorakis, Md Jalil Piran, and Thippa Reddy Gadekallu. Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, 55:102987, 2023.

[15] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[16] M.H. Rehman and M.M. Gaber. *Federated Learning Systems: Towards Next-Generation AI*. Studies in Computational Intelligence. Springer International Publishing, 2021. ISBN 9783030706036. URL https://books.google.dz/books?id=54gyzgEACAAJ.

[17] Kallista Bonawitz, Peter Kairouz, Brendan McMahan, and Daniel Ramage. Federated learning and privacy: Building privacy-preserving systems for machine learning and data science on decentralized data. *Queue*, 19(5):87–114, 2021.

[18] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society*, 1:45–61, 2020.

[19] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

[20] Dashan Gao, Ce Ju, Xiguang Wei, Yang Liu, Tianjian Chen, and Qiang Yang. Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography. *arXiv preprint arXiv:1909.05784*, 2019.

[21] Tongyan Wei, Ying Wang, and Wenjing Li. The deep flow inspection framework based on horizontal federated learning. In *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. IEEE, 2022.

[22] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. Vertical federated learning: Challenges, methodologies and experiments. *arXiv preprint arXiv:2202.04309*, 2022.

[23] Yusuf Efe. A vertical federated learning method for multi-institutional credit scoring: Mics. *arXiv preprint arXiv:2111.09038*, 2021.

[24] Jingtao Guo, Ivan Wang-Hei Ho, Yun Hou, and Zijian Li. Fedpos: A federated transfer learning framework for csi-based wi-fi indoor positioning. *IEEE Systems Journal*, 2023.

[25] Chao Huang, Jianwei Huang, and Xin Liu. Cross-silo federated learning: Challenges and opportunities. *arXiv preprint arXiv:2206.12949*, 2022.

[26] Wenti Yang, Naiyu Wang, Zhitao Guan, Longfei Wu, Xiaojiang Du, and Mohsen Guizani. A practical cross-device federated learning framework over 5g networks. *IEEE Wireless Communications*, 2022.

[27] Yafeng Lu, Xiong Huang, Yisheng Dai, Sabita Maharjan, and Yan Zhang. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. *IEEE Transactions on Industrial Informatics*, 14(8):3690–3700, 2018.

[28] Manjari Ganapathy. *An Introduction to Federated Learning and Its Analysis*. PhD thesis, University of Nevada, Las Vegas, 2021.

[29] Monik Raj Behera, Suresh Shetty, Robert Otter, et al. Federated learning using peer-to-peer network for decentralized orchestration of model weights. 2021.

[30] Souhila Badra Guendouzi, Samir Ouchani, and Mimoune Malki. Enhancing the aggregation of the federated learning for the industrial cyber physical systems. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 197–202. IEEE, 2022.

[31] Youyang Qu, Shiva Raj Pokhrel, Sahil Garg, Longxiang Gao, and Yong Xiang. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. *IEEE Transactions on Industrial Informatics*, 17(4):2964–2973, 2020.

[32] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[33] WANG Luping, WANG Wei, and LI Bo. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pages 954–964. IEEE, 2019.

[34] Xi Zhu, Junbo Wang, Wuhui Chen, and Kento Sato. Model compression and privacy preserving framework for federated learning. *Future Generation Computer Systems*, 140:376–389, 2023.

[35] Mahdi Beitollahi and Ning Lu. Flac: Federated learning with autoencoder compression and convergence guarantee. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 4589–4594. IEEE, 2022.

[36] Juncai Liu, Jessie Hui Wang, Chenghao Rong, Yuedong Xu, Tao Yu, and Jilong Wang. Fedpa: An adaptively partial model aggregation strategy in federated learning. *Computer Networks*, 199:108468, 2021.

[37] Zheyi Chen, Weixian Liao, Kun Hua, Chao Lu, and Wei Yu. Towards asynchronous federated learning for heterogeneous edge-powered internet of things. *Digital Communications and Networks*, 7(3):317–326, 2021.

[38] Hyun-Suk Lee and Jang-Won Lee. Adaptive transmission scheduling in wireless networks for asynchronous federated learning. *IEEE Journal on Selected Areas in Communications*, 39(12):3673–3687, 2021.

[39] Michael R Sprague, Amir Jalalirad, Marco Scavuzzo, Catalin Capota, Moritz Neun, Lyman Do, and Michael Kopp. Asynchronous federated learning for geospatial applications. In *ECML PKDD 2018 Workshops: DMLE 2018 and IoTStream 2018, Dublin, Ireland, September 10-14, 2018, Revised Selected Papers*, pages 21–28. Springer, 2019.

[40] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

[41] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 15–24. IEEE, 2020.

[42] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2019.

[43] Jiajun Wu, Steve Drew, and Jiayu Zhou. Fedle: Federated learning client selection with lifespan extension for edge iot networks. *arXiv preprint arXiv:2302.07305*, 2023.

[44] Osama Wehbi, Sarhad Arisdakessian, Omar Abdel Wahab, Hadi Otrok, Safa Otoum, Azzam Mourad, and Mohsen Guizani. Fedmint: Intelligent bilateral client selection in federated learning with newcomer iot devices. *arXiv preprint arXiv:2211.01805*, 2022.

[45] Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. Fault-tolerant federated reinforcement learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 34:1007–1021, 2021.

[46] José Ángel Morell and Enrique Alba. Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices. *Future Generation Computer Systems*, 133: 53–67, 2022.

[47] Lokendra Gour and Akhilesh A Waoo. Fault-tolerant framework with federated learning for reliable and robust distributed system. In *THEETAS 2022: Proceedings of The International Conference on Emerging Trends in Artificial Intelligence and Smart Systems, THEETAS 2022, 16-17 April 2022, Jabalpur, India*, page 219. European Alliance for Innovation, 2022.

[48] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[49] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

[50] Rituparna Saha, Sudip Misra, and Pallav Kumar Deb. Fogfl: Fog-assisted federated learning for resource-constrained iot devices. *IEEE Internet of Things Journal*, 8(10):8456–8463, 2020.

[51] Sannara Ek, François Portet, Philippe Lalanda, and German Vega. A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom) (PerCom 2021)*, Kassel, Germany, March 2021.

[52] Tarek Berghout, Toufik Bentrcia, Mohamed Amine Ferrag, and Mohamed Benbouzid. A heterogeneous federated transfer learning approach with extreme aggregation and speed. *Mathematics*, 10(19):3528, 2022.

[53] Pu Tian, Zheyi Chen, Wei Yu, and Weixian Liao. Towards asynchronous federated learning based threat detection: A dc-adam approach. *Computers & Security*, 108:102344, 2021.

[54] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[55] Chun-Han Yao, Boqing Gong, Hang Qi, Yin Cui, Yukun Zhu, and Ming-Hsuan Yang. Federated multi-target domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1424–1433, 2022.

[56] Yuwei Sun, Ng Chong, and Ochiai Hideya. Multi-source domain adaptation based on federated knowledge alignment. *arXiv preprint arXiv:2203.11635*, 2022.

[57] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[58] Yiying Li, Wei Zhou, Huaimin Wang, Haibo Mi, and Timothy M Hospedales. Fedh2l: Federated learning with model and statistical heterogeneity. *arXiv preprint arXiv:2101.11296*, 2021.

[59] Li Hu, Hongyang Yan, Lang Li, Zijie Pan, Xiaozhang Liu, and Zulong Zhang. Mhat: an efficient model-heterogenous aggregation training scheme for federated learning. *Information Sciences*, 560:493–503, 2021.

[60] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[61] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38. IEEE, 2017.

[62] Sameer Wagh, Divyesh Cao, Xiao Lu, and Pratik Jagtap. Falcon: Honest-majority maliciously secure framework for private deep learning. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1859–1876, 2020.

[63] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[64] Febrianti Wibawa, Ferhat Ozgur Catak, Murat Kuzlu, Salih Sarp, and Umit Cali. Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case. In *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, pages 85–90, 2022.

[65] Zizhen Liu, Si Chen, Jing Ye, Junfeng Fan, Huawei Li, and Xiaowei Li. Dhsa: efficient doubly homomorphic secure aggregation for cross-silo federated learning. *The Journal of Supercomputing*, 79(3):2819–2849, 2023.

[66] Pathum Chamikara Mahawaga Arachchige, Dongxi Liu, Seyit Camtepe, Surya Nepal, Marthie Grobler, Peter Bertok, and Ibrahim Khalil. Local differential privacy for federated learning. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages 195–216. Springer, 2022.

[67] Muah Kim, Onur Günlü, and Rafael F Schaefer. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2650–2654. IEEE, 2021.

[68] Min Yoon Kim, Jong-Hyouk Kim, Aitor Perez, and Roque A Calvo. Blockchain-empowered federated learning: Threats and countermeasures. *IEEE Access*, 7:76051–76064, 2019.

[69] Thomas Hardjono and Alex Pentland. Towards an interoperability architecture blockchain-enabled identity and access management. *IEEE Transactions on Engineering Management*, 2019.

[70] Min Chen, Yong Ma, Wenbin Li, Zhiqing Hao, and Weijia Lou. Distributed machine learning and blockchain for robust and secure iot-enabled smart cities. *IEEE Internet of Things Journal*, 2019.

[71] Tianyi Yang and Jian Mu. Ringfed: Reducing communication costs in federated learning. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2022. URL https://dl.acm.org/doi/abs/10.1145/3474085.3475409.

[72] Lin Wang, YongXin Guo, Tao Lin, and Xiaoying Tang. Client selection in nonconvex federated learning: Improved convergence analysis for optimal unbiased sampling strategy. *arXiv preprint arXiv:2205.13925*, 2022.

[73] Zhaoyang Li, Kai Zhou, Yuchen Zhang, Jun Liu, and Zeyan Li. Adaptive federated optimization under system heterogeneity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2021. URL https://dl.acm.org/doi/10.1145/3459637.3482215.

[74] Stephanie A Morey, Nicole A Thomas, and Jason S McCarley. Redundant-target processing is robust against changes to task load. *Cognitive Research: Principles and Implications*, 3 (1):1–17, 2018.

[75] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[76] Dashan Gao, Xin Yao, and Qiang Yang. A survey on heterogeneous federated learning. *arXiv preprint arXiv:2210.04505*, 2022.

[77] Artur Back de Luca, Guojun Zhang, Xi Chen, and Yaoliang Yu. Mitigating data heterogeneity in federated learning with data augmentation. *arXiv preprint arXiv:2206.09979*, 2022.

[78] Suraj Rajendran, Zhenxing Xu, Weishen Pan, Arnab Ghosh, and Fei Wang. Data heterogeneity in federated learning with electronic health records: Case studies of risk prediction for acute kidney injury and sepsis diseases in critical care. *PLOS Digital Health*, 2(3):e0000117, 2023.

[79] AuthorFirstName AuthorLastName. Techniques for measuring, understanding and overcoming the domain shift as a crucial step towards reliable use of deep learning in the future clinical pathology applications. *IEEE Journal of Biomedical and Health Informatics*, 25: 325–336, 2021. URL https://ieeexplore.ieee.org/document/9234592.

[80] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10071, 2022.

[81] Yun Hin Chan and Edith CH Ngai. Fedhe: Heterogeneous models and communication-efficient federated learning. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pages 207–214. IEEE, 2021.

[82] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[83] Aashma Uprety and Danda B Rawat. Mitigating poisoning attack in federated learning. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–07. IEEE, 2021.

[84] Florian Nuding and Rudolf Mayer. Data poisoning in sequential and parallel federated learning. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, pages 24–34, 2022.

[85] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems*, 34:994–1006, 2021.

[86] Jiahui Geng, Yongli Mou, Feifei Li, Qing Li, Oya Beyan, Stefan Decker, and Chunming Rong. Towards general deep leakage in federated learning. *arXiv preprint arXiv:2110.09074*, 2021.

[87] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Protect privacy from gradient leakage attack in federated learning. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 580–589. IEEE, 2022.

[88] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 797–807. IEEE, 2021.

[89] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021.

[90] Ali Hatamizadeh, Hongxu Yin, Pavlo Molchanov, Andriy Myronenko, Wenqi Li, Prerna Dogra, Andrew Feng, Mona G Flores, Jan Kautz, Daguang Xu, et al. Do gradient inversion attacks make federated learning unsafe? *IEEE Transactions on Medical Imaging*, 2023.

[91] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[92] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[93] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[94] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[95] Wikipedia. Homomorphic encryption.

[96] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[97] Cynthia Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5*, pages 1–19. Springer, 2008.

[98] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[99] Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022.

[100] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 24–43. Springer, 2010.

[101] Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic encryption and multiparty computation. In *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, pages 169–188. Springer, 2011.

[102] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[103] Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. The bounded laplace mechanism in differential privacy. *arXiv preprint arXiv:1808.10410*, 2018.

[104] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

[105] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: A comprehensive introduction. *Princeton University Press*, 2016.

[106] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. *Business & Information Systems Engineering*, 59:183–187, 2017.

[107] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016. URL http://arxiv.org/abs/1602.05629.

[108] Sannara Ek, François Portet, Philippe Lalanda, and German Vega. A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *19th IEEE International Conference on Pervasive Computing and Communications PerCom 2021*, 2021.

[109] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[110] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[111] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[112] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[113] Rui Ye, Zhenyang Ni, Chenxin Xu, Jianyu Wang, Siheng Chen, and Yonina C Eldar. Fedfm: Anchor-based feature matching for data heterogeneity in federated learning. *arXiv preprint arXiv:2210.07615*, 2022.

[114] Chamath Palihawadana, Nirmalie Wiratunga, Anjana Wijekoon, and Harsha Kalutarage. Fedsim: Similarity guided model aggregation for federated learning. *Neurocomputing*, 483: 432–445, 2022.

[115] Ian Jolliffe. Principal component analysis. *Wiley Online Library*, 58(2):303–305, 2002.

[116] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1: 281–297, 1967.

[117] Office home dataset. https://paperswithcode.com/dataset/o ice-hom. Accessed on May 20, 2023.

[118] Khandaker Mamun Ahmed, Ahmed Imteaj, and M Hadi Amini. Federated deep learning for heterogeneous edge computing. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1146–1152. IEEE, 2021.

[119] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.

[120] Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*, 1:177–186, 2010.

[121] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. *Nature*, 521(7553): 436–444, 2016.

[122] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[123] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[124] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[125] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

[126] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[127] Yi-Hsin Chen, Wei-Yu Lin, Chih-Chiang Hsu, and Yu-Chiang Frank Wang. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2011–2020, 2017.

[128] Wikitext-2 dataset. https://paperswithcode.com/dataset/wikitext-2. Accessed on May 20, 2023.

[129] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Towards developing a systematic approach to generate benchmark android malware datasets and classification. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–7. IEEE, 2018.

[130] N Koroniotis, N Moustafa, E Sitnikova, B Turnbull, and J Slay. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

[131] Jingjing Yao and Nirwan Ansari. Enhancing federated learning in fog-aided iot by cpu frequency and wireless power control. *IEEE Internet of Things Journal*, 8(5):3438–3445, 2020.

[132] Zirui Xu, Zhao Yang, Jinjun Xiong, Janlei Yang, and Xiang Chen. Elfish: Resource-aware federated learning on heterogeneous edge devices. *Ratio*, 2(r1):r2, 2019.

[133] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. Hierarchical federated learning across heterogeneous cellular networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8866–8870. IEEE, 2020.

[134] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[135] Heinrich von Stackelberg. Market structure and equilibrium. *Zeitschrift für Nationalökonomie*, 5(1-2):301–324, 1934.

[136] Tor Anderson, Chin-Yao Chang, and Sonia Martínez. Distributed approximate newton algorithms and weight design for constrained optimization. *Automatica*, 109:108538, 2019.

[137] Hong Liu, Shuaipeng Zhang, Pengfei Zhang, Xinqiang Zhou, Xuebin Shao, Geguang Pu, and Yan Zhang. Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. *IEEE Transactions on Vehicular Technology*, 70(6):6073–6084, 2021.

[138] School of Information University of California, Irvine and Computer Sciences. Kdd cup 1999 data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.

[139] Yang Li and Ananthram Swami. Deep learning for domain adaptation: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12):2664–2679, 2017.

[140] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[141] Rich Caruana. Multitask learning. In *Machine learning*, volume 28, pages 41–75. Springer, 1997.

[142] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[143] Piotr Przybyła, Piotr Lipiński, Simon Borchmann, and Janusz Tracz. Using shakespeare to evaluate dialogue models. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 301–305. Association for Computational Linguistics, 2018.

[144] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[145] Jian Ren, Xiaohui Shen, Zhe Lin, Radomir Mech, and David J Foran. Personalized image aesthetics. In *Proceedings of the IEEE international conference on computer vision*, pages 638–647, 2017.

[146] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[147] Hemanth Venkateswara, Jose Eusebio, and Shayok Chakraborty. Deep hashing network for efficient similarity retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1667–1675, 2017.

[148] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[149] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[150] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Association for Computational Linguistics, 2009. URL https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[151] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[152] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. Pushshift.io: A reddit dataset for political discourse analysis in the 2016 us presidential election. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 830–839, 2020.

[153] Google Creative Lab. Quick, Draw! https://quickdraw.withgoogle.com/data, 2016.

[154] Webank's AI Department. An industrial grade federated learning framework, 2019. URL https://fate.fedai.org/.

[155] Mathieu N Galtier and Camille Marini. Substra: a framework for privacy-preserving, traceable and collaborative machine learning. *arXiv preprint arXiv:1910.11567*, 2019.

[156] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[157] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, and D. Rueckert. Pysyft: A library for private, secure, and decentralized data science. https://github.com/OpenMined/PySyft, 2018.

[158] OpenMined. Pygrid: A peer-to-peer platform for private data science and federated learning. https://github.com/OpenMined/PyGrid, 2019.

[159] Alexandre Beutel, Paul Schmid, Rebecca Roelofs, Jared Dunnmon, Francis Li, Daniel Golovin, Sebastian Nowozin, Eric Turner, and Justin Solomon. Flower: A friendly federated learning research framework, 2020.

[160] Kwing Hei Li, Pedro Porto Buarque de Gusmão, Daniel J Beutel, and Nicholas D Lane. Secure aggregation for federated learning in flower. In *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*, pages 8–14, 2021.

[161] G Anthony Reina, Alexey Gruzdev, Patrick Foley, Olga Perepelkina, Mansi Sharma, Igor Davidyuk, Ilya Trushkin, Maksim Radionov, Aleksandr Mokrov, Dmitry Agapov, et al. Openfl: An open-source framework for federated learning. *arXiv preprint arXiv:2105.06413*, 2021.

[162] IBM. Ibm federated learning: An enterprise framework for collaborative machine learning. https://www.ibm.com/cloud/architecture/content/course/federated-learning, 2020.

[163] NVIDIA. Nvidia clara: Ai platform for healthcare and life sciences. https://developer.nvidia.com/clara, 2018.

[164] Wang J. Li Z. Wang Y. Liu K. He, H. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[165] Muhammad Asad, Ahmed Moustafa, Takayuki Ito, and Muhammad Aslam. Evaluating the communication efficiency in federated learning algorithms. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 552–557. IEEE, 2021.

[166] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. Measure contribution of participants in federated learning. In *2019 IEEE international conference on big data (Big Data)*, pages 2597–2604. IEEE, 2019.

[167] Xiaopeng Jiang and Cristian Borcea. Complement sparsification: Low-overhead model pruning for federated learning. 2023.

[168] In more depth: Mips, mops, and other flops. URL https://course.ccs.neu.edu/cs3650/ssl/TEXT-CD/Content/COD3e/InMoreDepth/IMD4-MIPS-MOPS-and-Other-FLOPS.pdf. Accessed: 2023-04-10.

[169] Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and S Yu Philip. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022.