Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.073519

REVIEW

A Survey of Federated Learning: Advances in Architecture, Synchronization, and Security Threats

Faisal Mahmud[1], **Fahim Mahmud**[2] **and Rashedur M. Rahman**[1,*]

[1]Department of Electrical and Computer Engineering, North South University, Bashundhara, Dhaka, 1229, Bangladesh
[2]Department of Computer Science and Engineering, Green University of Bangladesh, Purbachal American City, Kanchon, 1460, Bangladesh

*Corresponding Author: Rashedur M. Rahman. Email: rashedur.rahman@northsouth.edu

Received: 19 September 2025; Accepted: 21 November 2025; Published: 12 January 2026

ABSTRACT: Federated Learning (FL) has become a leading decentralized solution that enables multiple clients to train a model in a collaborative environment without directly sharing raw data, making it suitable for privacy-sensitive applications such as healthcare, finance, and smart systems. As the field continues to evolve, the research field has become more complex and scattered, covering different system designs, training methods, and privacy techniques. This survey is organized around the three core challenges: how the data is distributed, how models are synchronized, and how to defend against attacks. It provides a structured and up-to-date review of FL research from 2023 to 2025, offering a unified taxonomy that categorizes works by data distribution (Horizontal FL, Vertical FL, Federated Transfer Learning, and Personalized FL), training synchronization (synchronous and asynchronous FL), optimization strategies, and threat models (data leakage and poisoning attacks). In particular, we summarize the latest contributions in Vertical FL frameworks for secure multi-party learning, communication-efficient Horizontal FL, and domain-adaptive Federated Transfer Learning. Furthermore, we examine synchronization techniques addressing system heterogeneity, including straggler mitigation in synchronous FL and staleness management in asynchronous FL. The survey covers security threats in FL, such as gradient inversion, membership inference, and poisoning attacks, as well as their defense strategies that include privacy-preserving aggregation and anomaly detection. The paper concludes by outlining unresolved issues and highlighting challenges in handling personalized models, scalability, and real-world adoption.

KEYWORDS: Federated learning (FL); horizontal federated learning (HFL); vertical federated learning (VFL); federated transfer learning (FTL); personalized federated learning; synchronous federated learning (SFL); asynchronous federated learning (AFL); data leakage; poisoning attacks; privacy-preserving machine learning

1 Introduction

1.1 Motivation and Background

The rise of artificial intelligence and its integration into modern society have created transformative advancements in science, industry, and daily life. The success of machine learning (ML), particularly deep learning models, relies on their ability to learn complex patterns from large and diverse datasets to demonstrate capabilities in pattern recognition, prediction, and decision making [1]. These advances depend on processing of a vast amount of data and large-scale aggregation from domains such as medical diagnostics, financial transactions, and autonomous driving [2,3]. The practice of accumulating and centralizing large amounts of data, often containing deeply personal information such as financial transaction information,

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

location histories, and genomic sequences, has precipitated growing tension between technological progress and the fundamental right to privacy [4,5]. The tension is amplified by the high-profile data breaches and studies showing that even anonymized data can be re-identified with ease through a linkage that correlates de-identified datasets with publicly available information [6,7].

The growing privacy concern and rising public demand for data privacy and data sovereignty have led the government and regulatory bodies worldwide to enact data protection legislation. Landmark legislative frameworks, such as the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act (CCPA) in the United States, represent a paradigm shift in data governance with establishment of legal requirements for data handling such as data collection, processing, storage, data minimization (collecting only necessary data), purpose limitation (using data for the specified purpose for which it was collected), and granting individuals rights to access and erase their data [8]. These regulations have created an operational reality for the technology industry to make privacy preservation not merely an ethical consideration but a legal and commercial necessity, and increased the search for the technical dilemma of how to extract value from collective data without compromising the privacy of individuals [9].

Federated Learning (FL) has emerged as a promising decentralized machine learning paradigm by leveraging distributed datasets [10,11]. FL enables collaborative model training across multiple clients, such as mobile devices, hospitals, and financial institutions, to train a shared global model without the need for raw data in a central location. Clients train models on their private dataset locally and share parameter updates to the central server, which then aggregates these updates to refine the global model [12]. Data localization provides a built-in layer of privacy protection against server-side data breaches and external attackers. However, a significant amount of research has demonstrated that the shared model updates are information-rich and can be exploited by a malicious server to infer sensitive information about clients' private training data, whether a specific data point was part of the training set or not, or reconstruction attacks that aim to recover original training data from shared model updates [13,14].

Despite its promise, FL faces a cluster of practical and scientific challenges that slow its translation from controlled experiments to production systems. These challenges include multi-axis heterogeneity (data, model, and system), the unexpected privacy leakage conveyed by intermediate representations and model updates, large communication and computation costs on constrained devices, and vulnerabilities to poisoning and adaptive inference attacks. Synchronization and straggler management remain unresolved in large, volatile networks, and there is a persistent gap between theoretical guarantees and the behavior of realistic, compressed, or asynchronous protocols. Finally, the field suffers from fragmented evaluations and a lack of standardized benchmarks, which complicates reproducibility and the comparison of proposed defenses and optimizations [11,13,15]. These limitations motivate an integrative review that synthesizes recent advances, highlights cross-cutting trade-offs, and proposes a concrete research agenda to guide reliable, privacy-aware deployment of federated learning. This survey, therefore, synthesizes recent work into a coherent taxonomy, highlights the practical trade-offs, and draws attention to under-explored but high-impact issues such as benchmarking, reproducibility, and governance.

1.2 Scope of This Survey

Federated learning has matured rapidly, while its research directions have become fragmented. Many surveys focus narrowly on a single problem or on one application domain. Few synthesize recent advances across architectures, synchronization, and security together, especially work published between 2023 and 2025.

This survey fills that gap. It centers on three core dimensions that shape practical FL systems: architectural choices for data and model distribution, synchronization mechanisms for heterogeneous clients,

and privacy and security techniques for representation sharing. We give special attention to recent multi-party vertical FL, hybrid VFL-HFL frameworks, transformer-based personalization, and transfer methods that address sparse overlap. We also examine synchronization policies that trade stability for scalability and defenses that balance privacy, utility, and cost.

By keeping these dimensions in view at once, the survey reveals interactions that single-topic reviews miss. For example, personalization choices change privacy risk. Compression and client selection alter synchronization dynamics. Our scope covers methodological advances and their implications for deployment in healthcare, IoT, and edge systems.

1.3 Contributions of This Survey

This review offers a unified taxonomy that links architectures, synchronization strategies, and security mechanisms into a single analytical frame. That taxonomy makes it possible to compare methods along shared axes instead of treating each paper as an isolated solution.

We position this work against existing surveys by emphasizing cross-dimensional effects rather than narrow improvements. Where prior reviews catalog algorithms or applications, we highlight how heterogeneity, timing, and privacy interact and shape design trade-offs. We draw on 98 peer-reviewed papers from 2023 to 2025 to support our analysis. From that corpus, we extract recurring design patterns and note where progress is substantive and where fragmentation persists.

Finally, we deliver a focused research roadmap. Key directions include modular FL kernels that separate representation learning from task heads, adaptive privacy controllers that tune protection by measured leakage risk, standardized benchmarks and threat models, certified and layered defenses against adaptive attackers, and theoretical bounds for realistic protocols that combine compression, asynchrony, and privacy noise. We also call for operational toolkits for monitoring, debugging, and auditing that fit privacy constraints. Together, these contributions aim to guide research toward scalable, auditable, and deployable federated systems.

1.4 Structural Design of the Survey Work

Section 1 introduces the paper. It states the motivation and background in Section 1.1. It defines the scope of this survey in Section 1.2. It summarizes our main contributions in Section 1.3. It explains the structural design and how to navigate the review in Section 1.4.

Section 2 presents a comparative analysis of recent surveys. It highlights what prior reviews cover and what they miss. It explains where this survey adds value.

Section 3 develops our taxonomy of federated learning. The taxonomy links data distribution, training synchronization, and security threats.

Section 4 details the PRISMA-based methodology used for this review. It outlines the systematic search, screening process, and study selection criteria, which yielded the 98 studies included in the qualitative synthesis.

Section 5 presents the fundamentals of federated learning. Section 5.1 defines the core components: clients, the central aggregator, and the standard training workflow including local update, aggregation, and model dissemination. Section 5.2 reviews the primary FL architectures. It explains horizontal FL, vertical FL, federated transfer learning, and personalized FL, and it clarifies the assumptions and use cases that distinguish each paradigm. Section 5.3 contrasts synchronization strategies by comparing synchronous training with its straggler trade-offs and asynchronous alternatives with their staleness and convergence issues. Section 5.4 outlines key privacy and security concerns, focusing on representation and gradient

leakage, membership attack, property inference, poisoning, and backdoor attacks, and it frames the defensive primitives commonly applied.

Section 6 contains focused literature reviews organized by architecture. Section 6.1 examines vertical federated learning with three threads: handling heterogeneous participants, communication efficient and privacy enhanced protocols, and generative data augmentation techniques for low overlap scenarios. Section 6.2 examines horizontal FL by covering hybrid and heterogeneous designs, methods for communication optimization and feature selection, and application oriented frameworks that emphasize domain constraints. Section 6.3 surveys federated transfer learning, reviewing architectural proposals, privacy mechanisms, and concrete domain applications. Section 6.4 addresses personalized federated learning, detailing personalization mechanisms, fairness, and community structure, and cross-modal and domain-specific deployments.

Section 7 reviews synchronization strategies in depth. Section 7.1 surveys synchronous FL with straggler mitigation and semi-synchronous scheduling. Section 7.2 covers asynchronous FL with staleness compensation and scalable aggregation.

Section 8 reviews privacy and security research. Section 8.1 focuses on data leakage, including Gradient Attacks and Defenses, and on Privacy-Preserving Encodings. Section 8.2 examines poisoning attacks, backdoors, defense methods, and audit-style detection approaches.

Section 9 distills challenges and future directions. It organizes open problems into twelve items, from heterogeneity to sustainability. It ends with concrete research directions and a roadmap for researchers and practitioners.

Section 10 concludes by summarizing key findings and restating the survey contributions.

2 Comparative Analysis of Recent Surveys

To underscore this survey's distinctive contribution, we carry out a structured gap analysis that contrasts its scope, methodology, and findings with recent Federated Learning (FL) surveys from 2024–2025. The results indicate a persistent shift toward specialization in the literature, emphasizing the need for a unified perspective that examines interactions and trade-offs among core FL components.

Yurdem et al. (2024) [16] deliver a survey centered on FL software frameworks and tools, including TensorFlow Federated (TFF), PySyft, and Flower. They examine core principles, strategies, use cases, and available resources, while classifying FL variants (HFL/VFL) and addressing related issues like privacy and security in these tools. Although this provides a useful guide for developers choosing frameworks, the content remains largely catalog-like for current software options. As a result, it omits an in-depth, consolidated evaluation of the balances between key FL design elements (architecture, synchronization, security) that transcend particular framework details.

Vajrobol et al. (2025) [17] narrow their scope to FL applications in computational mental healthcare, surveying pertinent datasets and grouping use cases by mental health indicators such as depression, stress, and sleep patterns. They also cover various ML/DL frameworks in this clinical niche. While this domain-specific lens is noteworthy for healthcare experts, it inherently restricts broader insights into universal FL hurdles (e.g., general data heterogeneity or communication optimization outside healthcare) and omits a cohesive classification linking key FL aspects across varied fields.

In parallel, Wang et al. (2025) [18] examine FL tailored to Internet of Things (IoT) environments. They tackle privacy, threat detection, and communication efficiency in IoT setups with dense sensor networks and devices, detailing FL's role in such scenarios. Yet, this targeted perspective curtails its reach, omitting a wider

dissection of FL fundamentals beyond IoT and failing to integrate discoveries from disparate FL domains into a holistic model that evaluates architectural compromises.

Khanh et al. (2025) [19] hone in on edge-centric FL, especially for smart healthcare (IoHT). They assess AI-supported edge architectures and introduce an FL-driven IoHT framework aimed at minimizing latency and handling edge resource limitations. This confined emphasis on edge computing in healthcare diminishes its general relevance; the survey avoids a thorough breakdown of compromises pertinent to FL at large and skips a broad taxonomy extending past the edge/healthcare domain.

By comparison, Pei et al. (2024) [20] furnish a targeted and methodical assessment of heterogeneity (device, data, model) in FL. They probe the origins of heterogeneity and organize current approaches for these issues, yielding substantial insight into this domain. That said, their primary fixation on heterogeneity precludes a holistic dialogue on how such approaches influence other vital FL facets like security protections, synchronization options, or systemic resilience. The evaluation of key inter-pillar compromises is thus overlooked.

Nezhadsistani et al. (2025) [21] delve into Blockchain-augmented FL (BCFL), but confines itself to health care applications. They scrutinize BCFL elements like consensus mechanisms and encryption techniques, plus metrics for deployment and adherence to standards such as Health Insurance Portability and Accountability Act (HIPAA), GDPR in health care scenarios. This constrained intersection of blockchain and health care sharply limits its breadth, excluding a general FL viewpoint and a cohesive breakdown of architectural, synchronization, and security compromises applicable outside the BCFL health care specialty.

Lastly, Cai et al. (2024) [22] delve into blockchain-FL fusion (BC-FL). They evaluate advantages like enhanced decentralization and security, alongside drawbacks such as efficiency and storage issues, while organizing architectures and countermeasures. This blockchain-specific orientation means the survey forgoes a standalone FL overview and neglects a full examination of architecture-synchronization-security dynamics beyond the BC-FL paradigm.

These surveys show a clear tendency toward specialization. Some works highly concentrate on a single technical challenge, for example, heterogeneity [20], while others limit their scope to a single application domain, such as mental health [17]. Technology-centric reviews examine specific integrations such as blockchain with FL [21,22], or FL for IoT applications [18]. These focused studies provide depth and concrete solutions within their niches, but they also produce a fragmented picture of the field.

Across the literature, three recurring themes stand out. First, privacy and security remain the primary concerns and are treated either broadly [16] or as the central organizing principle in surveys of trustworthy FL [23]. Second, communication efficiency appears as a cross-cutting topic in many reviews, sometimes examined as an engineering constraint and sometimes framed as a component of trustworthiness [23]. Third, heterogeneity in data, system, and model dimensions receives varying levels of attention; Pei et al. provide one of the more systematic treatments of all three heterogeneity types [20].

Specialized surveys supply valuable, deep analyses, but they seldom analyze trade-offs across architectural, timing, and security choices. That omission leaves practitioners without guidance when a single deployment decision affects multiple objectives. Our survey addresses this gap by synthesizing advances from 2023 to 2025 and by explicitly analyzing interactions among architectures, synchronization strategies, and security mechanisms. Table 1 summarizes this comparative landscape and shows how prior surveys differ in focus and coverage.

Table 1: A comparative analysis of recent surveys in federated learning (2024–2025)

Survey (Year)	Primary focus/Contribution	Taxonomy focus	Review scope	Heterogeneity (Data/System/Model)	Synchronization strategies (Sync/Async/Partial)	Security & privacy	Communication efficiency	Integration with other tech	Application domains
Yurdem et al. (2024) [16]	FL software frameworks and tools (TFF, PySyft, Flower).	General/Foundational	HFL/VFL categorization.	✓/✓/X	X	✓	◐	✓ (BC/Edge)	General
Vajrobol et al. (2025) [17]	FL in computational mental healthcare.	Focused (Mental Health)	Apps by symptom.	◐/X/X	X	◐	X	X	Mental health
Wang et al. (2025) [18]	FL in IoT, privacy, attack detection, comm. efficiency.	Focused (IoT)	IoT applications.	✓/✓/◐	X	✓	✓	✓ (IoT)	IoT
Khanh et al. (2025) [19]	Edge-based FL for smart healthcare.	Focused (Healthcare/Edge)	IoHT Architecture.	◐/◐/X	X	◐	✓	✓ (Edge)	Healthcare
Pei et al. (2024) [20]	Challenges of heterogeneity, solutions.	Focused (Heterogeneity)	Causes and solutions.	✓/✓/✓	◐	X	✓	◐ (5G/6G)	Consumer electronics
Nezhadsistani et al. (2025) [21]	Blockchain-enabled FL in healthcare.	Focused (BC/Healthcare)	BCFL with compliance.	✓/◐/X	X	✓	◐	✓ (IoMT)	Healthcare
Cai et al. (2024) [22]	Blockchain integration with FL.	Focused (Blockchain)	BC-FL architectures.	X/◐/X	◐	✓	✓	✓ (BC)	General
Tariq et al. (2024) [23]	Trustworthy FL: fairness, accountability, comm. efficiency.	General (Trust)	Trust pillars.	✓/✓/◐	◐	✓	✓	◐ (BC)	General
Our Survey (2025)	Unified review on architecture, synchronization, security, & trade-offs.	General/Foundational	Architecture, sync, security.	✓/✓/✓	✓	✓	✓	✓ (IoT/BC/Edge)	General

Note: Legend ✓ = Comprehensive Coverage ◐ = Partial Coverage X = No Significant Focus.

In short, specialized reviews deliver rigorous treatments of important subproblems. A unified review is still needed to connect those treatments into a practical design space. This article aims to provide that integration and a practical roadmap for researchers and system builders working toward deployable, trustworthy federated learning systems.

3 Taxonomy of Federated Learning

The architecture, complexity, and applicable use cases of a federated learning system are dictated by the data partition across participating entities or clients. The categorization of reviewed papers is based on three fundamental questions: how the data is split, how models are synchronized, and how to defend against attacks. The FL systems are categorized into four primary architectures based on their distribution

of features and samples, which are identified as Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), Federated Transfer Learning (FTL), and Personalized FL. Based on how model update synchronization is performed, the FL system is categorized into Synchronous and Asynchronous FL, and considering the vulnerabilities, the papers are divided into two categories: Deep leakage-based vulnerabilities and Data Poisoning-based vulnerabilities. This architectural choice mostly shapes the nature of the security and privacy challenges a system will face, along with determining the primary attack surfaces and the corresponding defense strategies required to prevent those attacks.

While classic surveys such as Kairouz et al. [15] organized federated learning around algorithmic families, problem settings, and privacy primitives, our survey departs from that framing in three concrete ways. First, we elevate training synchronization to a primary design axis alongside data distribution and security so that timing and staleness are treated as explicit variables that affect algorithm choice and risk. Second, we analyze cross-dimensional interactions rather than catalog techniques independently, showing how an architectural decision reshapes privacy leakage, communication cost, and robustness. Third, we extend the empirical window to cover 2023 through 2025 and thereby capture recent innovations such as prototype aggregation, hybrid VFL HFL architectures, transformer-based personalization, and adaptive privacy controllers. Together, these elements form a practical design map that links research choices to deployment trade-offs and evaluation criteria.

This taxonomy demonstrates a unified perspective by synthesizing and organizing recent specialized works into a coherent structure. Rather than presenting parallel surveys on FL for Healthcare (Vajrobol et al., 2025 [17] and Khanh et al., 2025 [19]), FL for IoT (Wang et al., 2025 [18]), Trustworthy FL (Tariq et al., 2024 [23]), or FL Heterogeneity (Pei et al., 2024 [20]), the three-axis model provides a common foundation that classifies each along Architecture, Synchronization, and Security. For example, an IoT study (Wang et al., 2025 [18]) is positioned under the Architecture axis as an application domain with synchronization needs that are often asynchronous due to device heterogeneity and with security threats that reflect IoT attack surfaces. A survey on Trustworthy FL (Tariq et al., 2024 [23]) fits as a deep analysis within the Security and Privacy axis. This unified approach encourages practitioners to view these areas as interconnected components of a single design space and to assess the trade-offs that arise across axes.

This unified perspective is essential for practice because earlier taxonomies that emphasize only data partitioning or security often relegate synchronization to an implementation detail or a subproblem. This is evident in the recent literature. Application-specific surveys, such as those on IoT (Wang et al., 2025 [18]) and healthcare (Khanh et al., 2025 and Vajrobol et al., 2025 [17,19]), identify low latency and communication efficiency as critical challenges within their domains. Problem-specific surveys on heterogeneity (Pei et al., 2024 [20]) present device heterogeneity and stragglers as primary problems and list asynchronous interaction as a future research direction. Security-focused surveys on Trustworthy FL (Tariq et al., 2024 [23]) and Blockchain-enabled FL (Cai et al., 2024 and Nezhadsistani et al., 2025 [21,22]) include communication efficiency and consensus within broader pillars of trustworthiness and scalability. General framework overviews such as Yurdem et al. (2024 [16]) describe system and data heterogeneity and communication overloads as distinct challenges rather than as a primary organizing principle. While these perspectives are valuable, they can obscure a fundamental high-level design trade-off. This taxonomy elevates synchronization because the choice between Synchronous FL and Asynchronous FL is an early and consequential decision. It is not only a remedy for heterogeneity. It is a primary architectural choice that governs core system behavior and interacts in a cascading manner with the other two axes.

Our organizing principle directs practitioners to a clear decision between two paradigms with non-negotiable costs. For example, Synchronous FL is appropriate when the primary goals are model stability and simpler security integration, since secure aggregation protocols are straightforward to implement in

lockstep. The corresponding cost is the straggler problem, where overall training speed is limited by the slowest client. Asynchronous FL is appropriate when the priority is higher throughput and scalability, since it removes the straggler bottleneck. The associated cost is model staleness, which arises from the use of outdated gradients and complicates convergence analysis, can degrade model accuracy, and increases the complexity of secure aggregation.

This perspective supports informed and deliberate system-level design. By presenting Architecture, Synchronization, and Security as the primary interacting design choices, the taxonomy provides a complete map for practice. These are not independent problems to be solved one by one. A decision on one axis directly influences the others. For example, selecting an asynchronous synchronization model to improve scalability complicates the security model for secure aggregation and may encourage architectural personalization to manage divergence caused by staleness. The framework, therefore, promotes holistic reasoning about how interdependent design choices shape outcomes, rather than addressing heterogeneity or security in isolation.

Fig. 1 illustrates the proposed taxonomy of Federated Learning (FL) systems, classifying them according to three key aspects. These categories include the method of data distribution (e.g., Horizontal, Vertical, Transfer FL, and Personalized FL), the model synchronization strategy (Synchronous vs. Asynchronous), and major security considerations such as data leakage and data poisoning vulnerabilities.

4 Methodology and Literature Selection

4.1 Overview

We adopted a Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) 2020-based methodology to identify and curate recent research on federated learning (FL) for the purposes of validating a unified taxonomy. The PRISMA-adapted approach was refined to prioritize top-tier publications and contributions from leading research institutions. The overall workflow combined systematic retrieval, automated and manual deduplication, dual-reviewer title/abstract screening, and independent full-text eligibility assessment across multiple scholarly repositories. The identification stage is summarized in Table 2. Searches were executed to target records published between January 2023 and October 2025.

4.2 Information Sources and Search Strategy

To balance domain specificity and breadth, we applied a two-tier search approach. Primary indexed sources (IEEE Xplore, ACM Digital Library, and Scopus) were queried to capture peer-reviewed computing and engineering literature. Supplementary sources (Google Scholar, arXiv, and PubMed) were searched to ensure coverage of cross-disciplinary studies and applications of FL. The search strategy combined core terms (e.g., “federated learning”, “decentralized learning”) with topic-specific keywords corresponding to the review’s taxonomy. Logical operators, phrase searches, and year filters were applied.

The topic-specific queries were constructed using the following taxonomy topics: Vertical Federated Learning, Poisoning Attacks, Jamming Attacks, Asynchronous FL, Synchronous FL, Heterogeneous FL, Personalized FL, Federated Transfer Learning, Gradient Leakage, Privacy-Preserving FL, Differential Privacy, and Hybrid FL.

4.3 Study Selection and Eligibility Criteria

All records retrieved from the five repositories were exported into a centralized screening database and deduplicated using automated matching (title, DOI, author list), followed by manual inspection for near-duplicates. Screening proceeded in two stages:

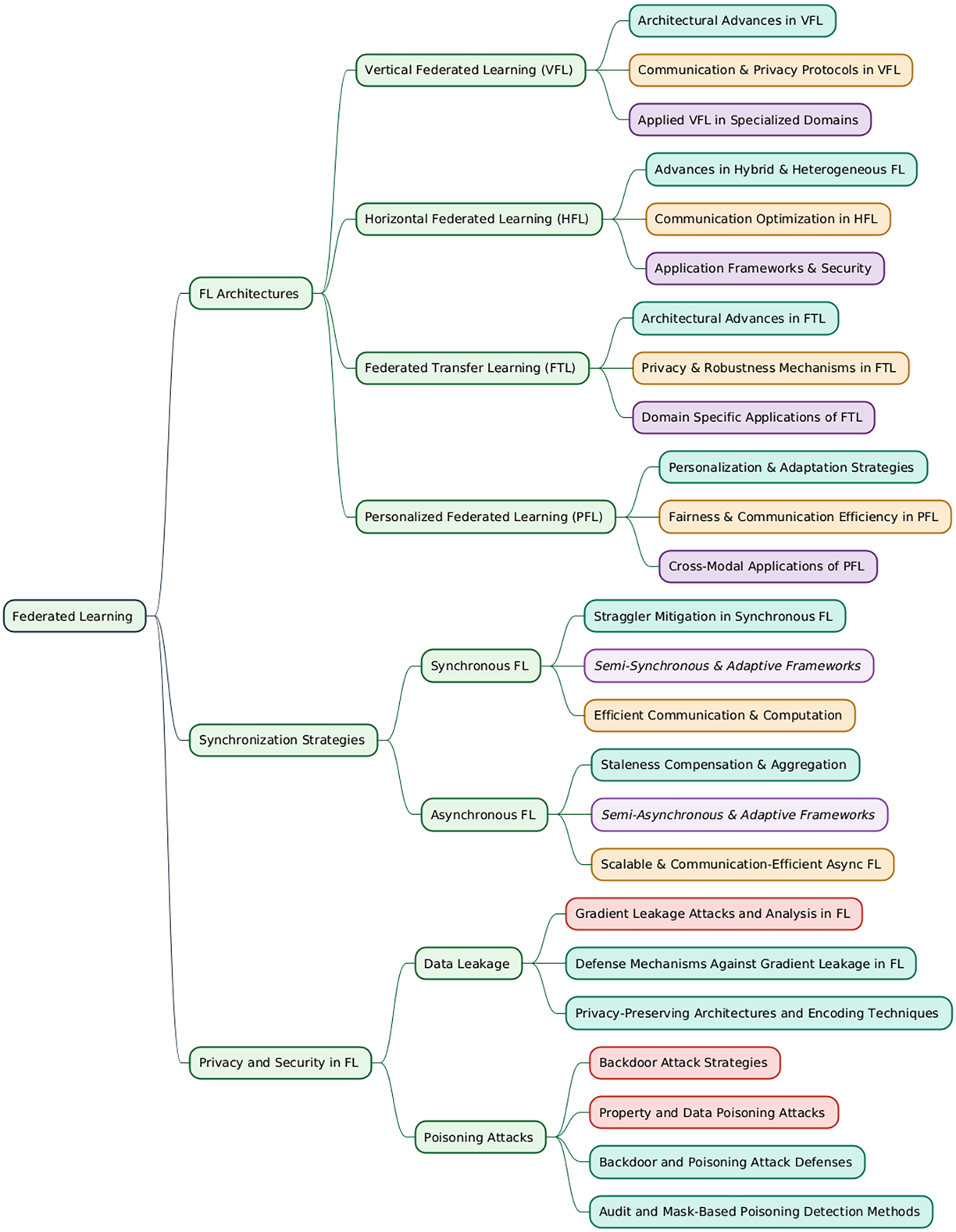

Figure 1: A taxonomy of federated learning systems

Table 2: PRISMA summary of study selection (federated learning literature, 2023–2025)

PRISMA stage	Count	% of previous stage	Notes
Records identified through database searching	9812	100.0%	Google Scholar (4322); IEEE Xplore (1779); ACM DL (1271); arXiv (2034); PubMed (406)
Duplicates removed	4727	48.18% (of identified)	Overlap across databases
Records after duplicates removed (unique)	5085	51.82% (of identified)	Retained for screening
Records excluded (title/abstract screening)	4415	86.82% (of after dedup)	Irrelevant/out-of-scope
Full-text articles assessed for eligibility	670	13.18% (of after dedup)	Retrieved for detailed review
Full-text articles excluded	572	85.37% (of assessed)	Wrong methodology, insufficient results, unavailable full text, duplicates at full-text stage
Studies included in qualitative synthesis	98	14.63% (of assessed)	Final curated corpus (1.93% of unique records)

Title and Abstract Screening.

Two independent reviewers screened titles and abstracts for topical relevance and compliance with pre-specified inclusion and exclusion criteria. Inclusion required original research (methods, experiments, or empirical evaluations) relevant to FL and one or more taxonomy topics. Exclusion criteria included non-research items (editorials, commentaries), works not focused on FL, or papers lacking sufficient methodological detail. Conflicts were resolved through discussion.

Full-Text Eligibility Assessment.

Full texts were retrieved for all records that passed title/abstract screening. Two reviewers independently assessed each full text against the inclusion/exclusion criteria and recorded a primary reason for exclusion where applicable.

4.4 Data Extraction and Synthesis

From the included studies, we extracted bibliographic metadata, research objectives, FL setting (e.g., cross-device vs cross-silo, vertical vs horizontal), threat model or challenge addressed, proposed methods or defenses, datasets, experimental setup, evaluation metrics, and key empirical results. Extracted items were organized to support taxonomy validation and to identify trends, gaps, and open problems. We used narrative synthesis to integrate methods and findings, supplemented by quantitative summaries (counts, timelines, topic prevalence) where appropriate.

4.5 Search Results and Selection

The combined search returned 9812 records (Google Scholar 4322; IEEE Xplore 1779; ACM Digital Library 1271; arXiv 2034; PubMed 406). After automated and manual deduplication, 4727 duplicates were

removed, yielding 5085 unique records for title/abstract screening. Title/abstract screening excluded 4415 records (86.82% of screened), leaving 670 full texts for retrieval and eligibility assessment. Following full-text review, 572 articles were excluded for pre-specified reasons (wrong methodology, insufficient results, unavailable full text, or duplicates identified at full-text stage), producing a final qualitative synthesis of 98 studies.

All counts and percentages reported in Table 2 are internally consistent and derived from the stage totals above.

4.6 Meta-Analysis of Relevant Literature

Fig. 2 combines two complementary views of the search results and topic distribution. Fig. 2a shows the initial retrieval across five repositories (n = 9812), with Google Scholar yielding the largest share (4322), and IEEE Xplore (1779), arXiv (2034), and ACM Digital Library (1271) contributing substantial domain-specific coverage. Fig. 2b shows the 5085 unique papers mapped to the review's taxonomy topics: Privacy-Preserving FL dominates (1050), followed by Personalized FL (725) and Heterogeneous FL (600). Less represented topics include Hybrid FL (210) and Jamming Attacks (170). This two-panel map contextualizes the broader landscape from which the final n = 98 studies were systematically selected.

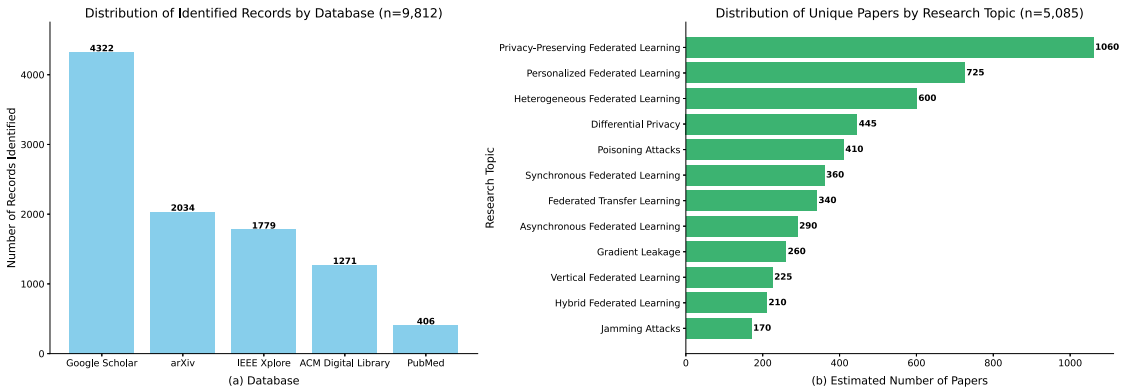

Figure 2: Figure shows summary of literature sources and topic prevalence. Fig. 2a shows the distribution of 9812 identified records across five databases. Fig. 2b displays the distribution of 5085 unique papers across taxonomy topics

Fig. 3 combines publication-type and temporal-distribution views of the final corpus. Fig. 3a shows the distribution of publication types, with journals comprising roughly 80%–85% of included studies and conferences approximately 15%–20%, reflecting the field's maturation and emphasis on reproducible, archival work. Fig. 3b shows the temporal distribution across 2023–2025: research activity peaked in 2024 (constituting more than half of the corpus), while 2023 and 2025 each contribute smaller but meaningful shares. Together, these panels illustrate both the venue profile and the recent surge in FL research that motivated the present taxonomy and gap analysis.

Fig. 4 displays the venue quality distribution of the included works, categorized by journal quartiles (Q1–Q4) and conference rankings (A, Others). A majority of journal publications fall within Q1 and Q2 quartiles, underscoring their concentration in leading outlets such as IEEE Trans. Neural Networks Learn. Syst., Information Fusion, and ACM Computing Surveys. Conference contributions primarily originate from A-ranked venues (e.g., NeurIPS, ICML, AAAI), with a small proportion from unranked or regional events. This pattern confirms the overall quality and credibility of the selected corpus. Taken together, these findings

highlight a maturing research ecosystem in federated learning that substantiates the corpus as the basis for taxonomy validation and gap analysis.

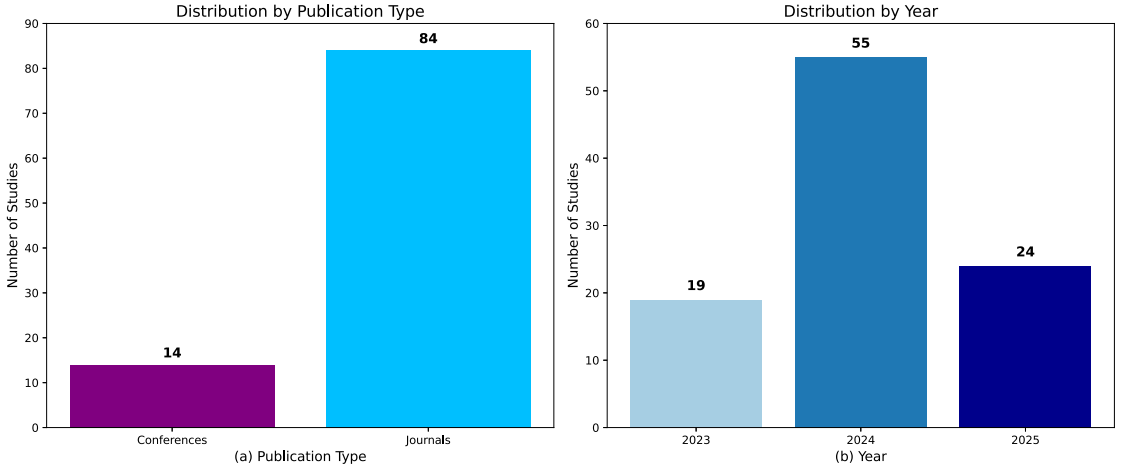

Figure 3: Distribution of the 98 included studies by publication type and year. Fig. 3a shows the breakdown by publication venue, where journals account for 84 studies (≈86%) and conferences for 14 (≈14%). Fig. 3b displays the temporal distribution of publications

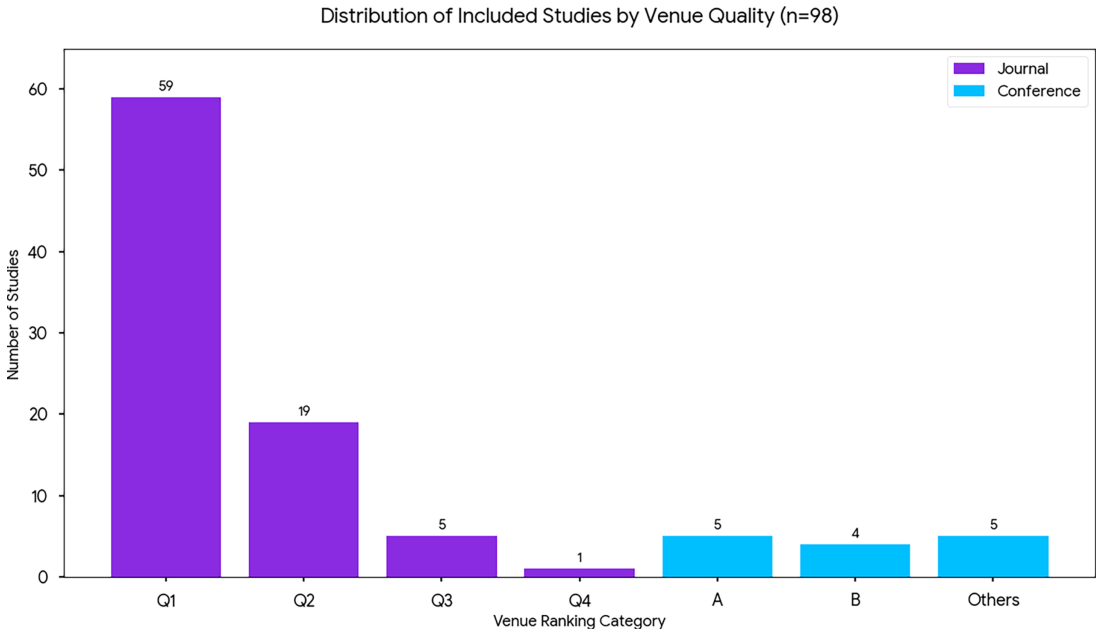

Figure 4: Venue quality distribution by journal quartile (Q1–Q4) and conference ranking

5 Fundamentals of Federated Learning (FL)

Federated Learning is a machine learning technique that enables multiple participating clients to collaboratively train a shared global model without exchanging their raw, sensitive local data with the central server [10]. This decentralized training approach is driven by the imperative to maintain data privacy and ownership in scenarios involving highly regulated or proprietary datasets, where data centralization is impractical or undesirable [24]. FL focuses on aggregating computation instead of data aggregation, where

individual clients perform local model updates on their private datasets, and only the model updates are shared and aggregated at a central server [10]. This architectural design offers stronger privacy compared to traditional centralized learning methods, where all training data resides in one single location, making it a single point of vulnerability open to various attacks and potential regulatory failures [25]. The collaborative learning process enables the global model to benefit from the collective learning of all participating clients, which allows the global model to generalize from a richer and more diverse dataset than any single client possesses, all while keeping data private and secure in local servers. This is particularly crucial in domains that run under a regulatory framework such as healthcare, where patient data often cannot legally leave the hospital where it was generated to keep patients' sensitive data protected [26,27]. The core principles of FL revolve around model refinement through an iterative process that involves two key components: Clients and a Central Server, which works as an Aggregator [15,28,29].

5.1 Clients

In the FL architecture, clients are distributed entities that hold the training data and possess the computational resources to train local models. These are the endpoints that hold the raw, often sensitive, training data and are also equipped with the necessary computational power to perform model training locally. Kairouz et al. (2021) [15] highlight a fundamental and critical distinction within the field, which categorizes FL into two primary standards based on the characteristics of these clients: Cross-Device Federated Learning and Cross-Silo Federated Learning.

Cross-Device Federated Learning: Training at Massive Scale The Cross-Device FL involves a massive number of end-user devices [30]. These clients are devices that consist of smartphones, tablets, Internet of Things (IoT) gadgets, or personal computers, each with a relatively small amount of data. The scale of this setting may reach millions, or even billions, of participating users and devices [31].

Client & Data Characteristics Each client in the Cross-Device FL environment holds a relatively small dataset that reflects the direct interaction of a single user. This data is inherently non-identically and independently distributed (non-IID) across the network and captures unique personal characteristics, behaviors, preferences, and environments [32].

Environmental Constraints Individual devices in Cross-Device FL have limited resources, such as low computational power, memory, and battery life, which limits the complexity of the models that can be trained locally. Clients often connect over unreliable or low-bandwidth networks such as cellular data and public Wi-Fi, which makes frequent and large model updates impractical [33]. The user base of available clients is highly dynamic. Devices may participate opportunistically and can drop out unexpectedly due to network issues, user behavior, or low battery. These issues pose a significant challenge to the synchronization and convergence of the global model [34].

Cross-Silo Federated Learning: Collaborative Institutional Training

The Cross-Silo FL setting involves a smaller, stable, and powerful cohort of clients. These clients are not individual devices but rather institutional entities, data centers, or organizations, often referred to as "silos" that possess their own dataset but are unwilling or unable to share the data directly [35]. The number of participants ranges from two to around one hundred. The purpose of this kind of collaboration is to include hospitals collaborating on medical research, financial institutions developing fraud detection models, or independent research labs combining their findings.

Client & Data Characteristics

Each silo represents an individual organization or institution that holds a large and generally high-quality dataset collected from its own users, systems, or operations. While these datasets are rich enough to

train effective local models, they have a tendency to reflect only the specific characteristics of that particular institution, such as a hospital's patient demographics or a company's customer base [36]. As a result, models trained in isolation suffer from limited generalizability. Cross-Silo Federated Learning enables multiple silos to collaboratively train and create a shared global model without sharing their raw data. The global model benefits from a broader, more diverse range of data than any single silo could have achieved if it had trained on its own local data only [37].

5.2 Central Server: Aggregator

Most federated learning setups consist of a central server that serves as the coordinator and aggregator of the training process. Its main responsibilities include initialization of the global model, selection of a subset of clients for each training round, collection of model updates from those clients, and aggregation of gradient updates to create and refine the global model, which is then sent back to the clients in an iterative process. In many threat models, the central server is considered honest-but-curious. It follows the protocol correctly but may try to infer sensitive information from the model updates of clients it receives [14]. The level of trust placed in the server influences the strength and type of privacy mechanisms that need to be employed, such as secure aggregation or differential privacy.

5.3 Standard Federated Learning Workflow

The foundational algorithm behind most federated learning (FL) implementations is Federated Averaging (FedAvg), introduced by McMahan et al. in 2017 [10]. The FedAvg algorithm combines local stochastic gradient descent (SGD) on client devices with an iterative averaging of model updates on the central server. This process serves as the foundational framework for training a single global model across decentralized clients without requiring them to share their local data. A standard FL process unfolds in iterative communication rounds, with each round generally consisting of four key stages:

Initialization, Client Selection and Model Distribution: The central server starts a round by selecting a random subset of its available clients to participate in the current round and sends the current state of the global model to each of the selected clients.

Local Training: After receiving the global model, each selected client initializes its local model and performs one or more steps of an optimization algorithm, typically Stochastic Gradient Descent (SGD), on its local dataset. The number of local training epochs is a crucial hyperparameter. While performing more local epochs can reduce the number of communication rounds required for the global model convergence, this may lead clients to overfit their local data and diverge from the shared global objective, a phenomenon referred to as "client drift" [38]. To solve this issue, algorithms such as SCAFFOLD [39] have been proposed, which use control variates to track and compensate for the difference between the local and global models.

Update Communication: Each client sends its computed model update, which is communicated to the server as the difference between the updated local weights (new) and the initial global weights back to the central server.

Aggregation: The server waits to receive updates from a predetermined number of participating clients. Once updates received from a number of clients reach a specific threshold, the server aggregates these updates to construct the new global model. In the FedAvg algorithm, this aggregation is a weighted average of the client parameters, where each client's update is weighted by the number of data points in its local dataset.

This weighting method ensures that clients who trained on more local data have a proportionally greater influence on the final global model. Once aggregation is computed, the process repeats with the new global

model sent to the client in the next round. This iterative procedure of sharing gradients continues until the model converges on a held-out validation set or reaches a predefined number of rounds.

Fig. 5 shows the iterative process of the FedAvg algorithm. The process begins with Step 1, where a central server distributes a global model to multiple clients. In Step 2, each client independently trains this model on its local data, which never leaves the device. Following local training, clients send their model updates back to the server in Step 3. Finally, in Step 4, the server aggregates these individual updates to produce an improved global model for the next iteration. This entire process ensures data privacy by sharing only model updates, not the raw client data.

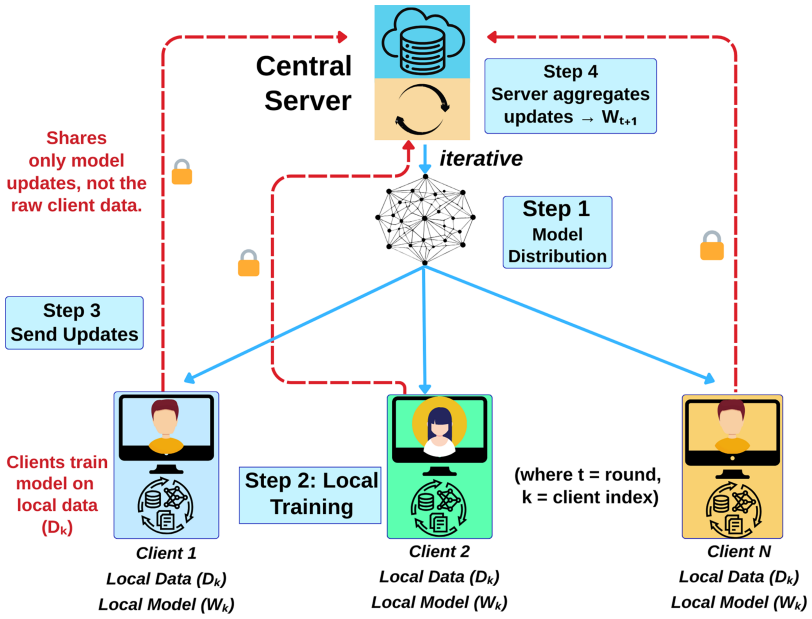

Figure 5: The Federated Averaging (FedAvg) learning process

5.4 Federated Learning Architectures

The FL architectures can be categorized into four categories, which are Horizontal Federated Learning, Vertical Federated Learning, Federated Transfer Learning, and Personalized Federated Learning.

Fig. 6 illustrates a classification of Federated Learning (FL) architectures based on how data is distributed among clients. The categories include Horizontal FL, where clients share the same feature space but have different data samples, Vertical FL, where clients have different features for the same set of samples, Federated Transfer Learning, applied when clients differ in both their features and samples, and Personalized FL, which addresses data heterogeneity by training models personalized to individual client objectives.

5.4.1 Horizontal Federated Learning (HFL)

Horizontal Federated Learning (HFL) is an approach where participants train models on local data that shares the same features but has different data instances (samples), which are often non-identically distributed (non-IID) [40–42]. HFL generally follows an iterative procedure where a central server aggregates model parameters from clients, who train locally on private data [43,44]. While raw data never leaves the client, ensuring inherent privacy [45], security risks like inference and model poisoning attacks remain [46]. Additional mechanisms like secure aggregation are used to mitigate these threats.

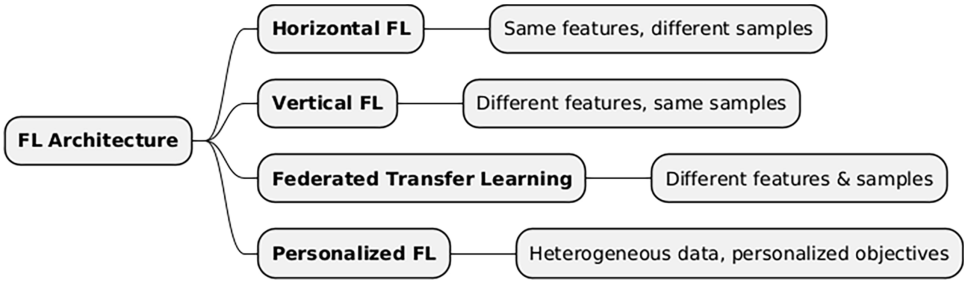

Figure 6: Classification of federated learning architectures

Fig. 7 illustrates Horizontal Federated Learning (HFL), where multiple hospitals hold the same types of data (e.g., patient age, blood pressure) for their respective patients. HFL is widely used in domains like healthcare, finance, and mobile computing.

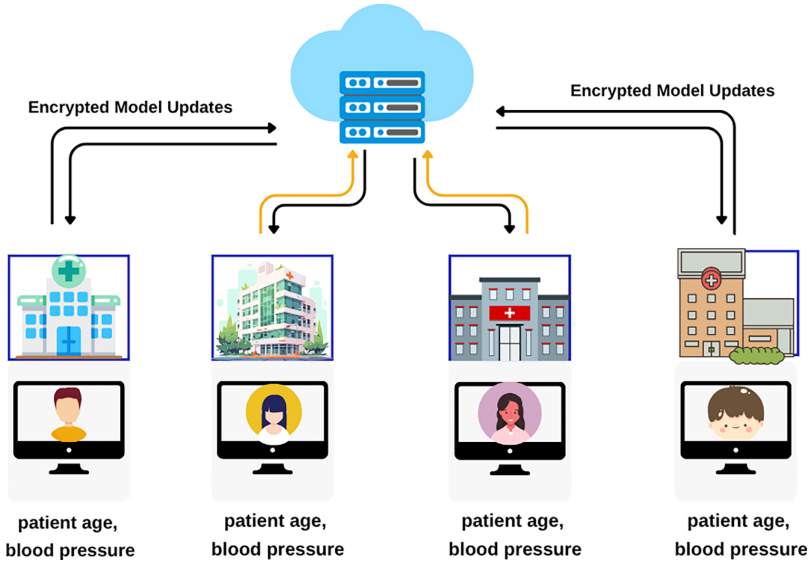

Figure 7: Horizontal federated learning

5.4.2 Vertical Federated Learning (VFL)

Vertical Federated Learning (VFL) applies when participants have datasets with the same samples (e.g., users) but different features [47]. For instance, a bank holds a user's financial features while an e-commerce site holds their behavioral features. Parties first securely identify common samples (e.g., via private set intersection [48]) and then collaboratively train a model. Each party trains a portion of the model on its unique features, and a coordinator combines intermediate results to update the global model. This leverages combined features for a more powerful model than any single party could build [49,50].

Fig. 8 illustrates the concept of Vertical Federated Learning (VFL). This paradigm is applied when different organizations hold different features for the same set of users. As shown, a bank may have a user's financial data (e.g., credit score), a hospital holds their medical records (e.g., diagnosis history), and a grocery store has their purchase history. In VFL, these entities collaborate to train a comprehensive model by jointly computing model updates using their distinct features, coordinated by a central server, without revealing their private data to each other.

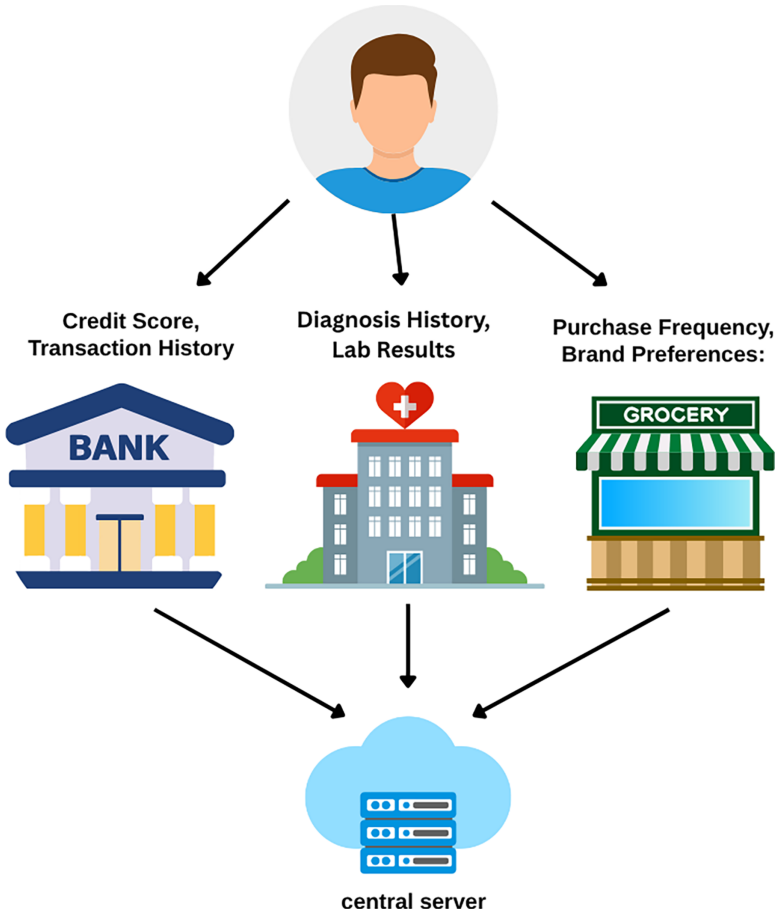

Figure 8: Vertical federated learning

VFL is not immune to risks like inferring private features from shared gradients [51–53], so it is often integrated with techniques like homomorphic encryption (HE) or differential privacy (DP). VFL also faces challenges with computational efficiency, communication overhead, and convergence. VFL architectures are well-suited for cross-sector collaboration, such as in financial services, healthcare, and marketing [54,55].

5.4.3 Federated Transfer Learning (FTL)

Federated Transfer Learning (FTL) combines federated and transfer learning to train models across data silos with different features and different samples [56]. Unlike HFL or VFL, FTL handles scenarios with little to no overlap in features or samples, making it suitable for cross-domain organizations with high data heterogeneity [57]. In FTL, parties with unique, non-overlapping datasets (e.g., health wearables and banking apps) improve local models by transferring knowledge from other domains [58,59]. This relies on transferring knowledge via a shared representation [60]. A central server aggregates intermediate results (like hidden layer representations) to distill generalizable patterns into a shared model.

Fig. 9 illustrates Federated Transfer Learning (FTL), used when participants have different tasks and feature sets, as with the medical models shown. Each model combines a local "Task Specific Head" with a "Shared Encoder." Gradients from the shared encoder are sent to a server, which aggregates them to create an updated, shared representation.

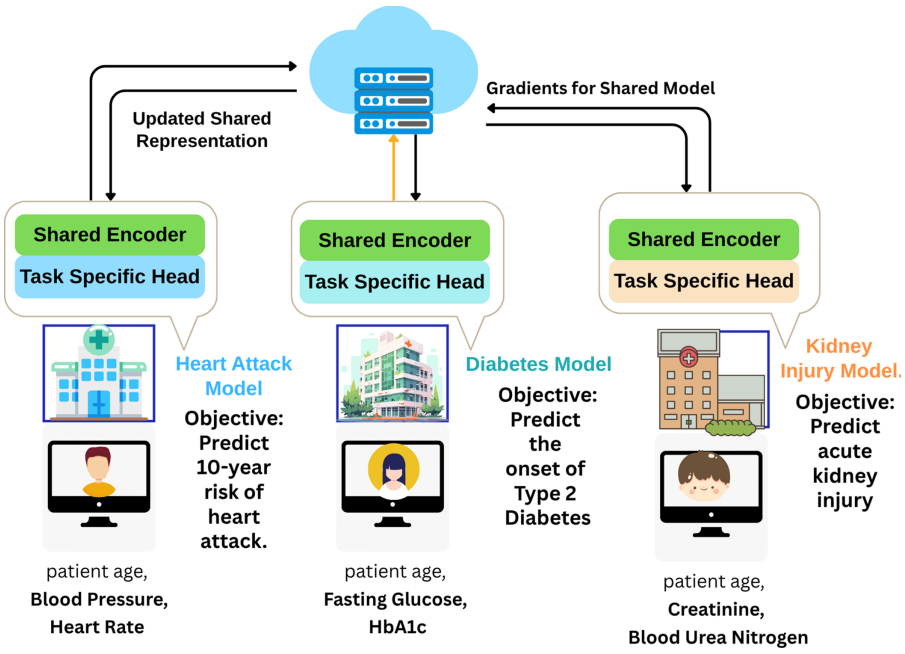

Figure 9: Federated transfer learning in healthcare

FTL architectures often incorporate domain adaptation, knowledge distillation, and cross-domain representation learning [57,61,62]. To minimize privacy risk, privacy-preserving mechanisms like secure aggregation, differential privacy, and homomorphic encryption are often integrated [56,63,64].

5.4.4 Personalized Federated Learning (PFL)

Personalized Federated Learning (PFL) addresses statistical heterogeneity, as the standard "one-size-fits-all" global model can be suboptimal for clients with non-independent and identically distributed (non-IID) data [65]. PFL enables each client to train a personalized model that combines collective insights with client-specific data, useful for tasks like keyboard prediction where user data varies widely [66].

One common PFL method divides parameters into shared global parameters (for general trends) and private local parameters (for unique characteristics) [67,68]. Other PFL techniques include regularization-based methods like FedProx, which penalizes divergence from the global model [69], clustering clients with similar data [70], and meta-learning for rapid adaptation [71].

Implementing PFL is challenging, demanding more computation and communication than standard FL [72], and requires robust privacy-preserving protocols [73].

Fig. 10 illustrates Personalized Federated Learning (PFL) applied to healthcare. A global model is trained using data from diverse clients (e.g., an athlete, a sedentary user, and a patient with cardiac anomalies). The model is then personalized on each user's local, non-IID electrocardiogram (ECG) data to generate specialized models without sharing sensitive health information.

Table 3 summarizes the federated learning architectures along with their data distributions, key security issues, and common threats.

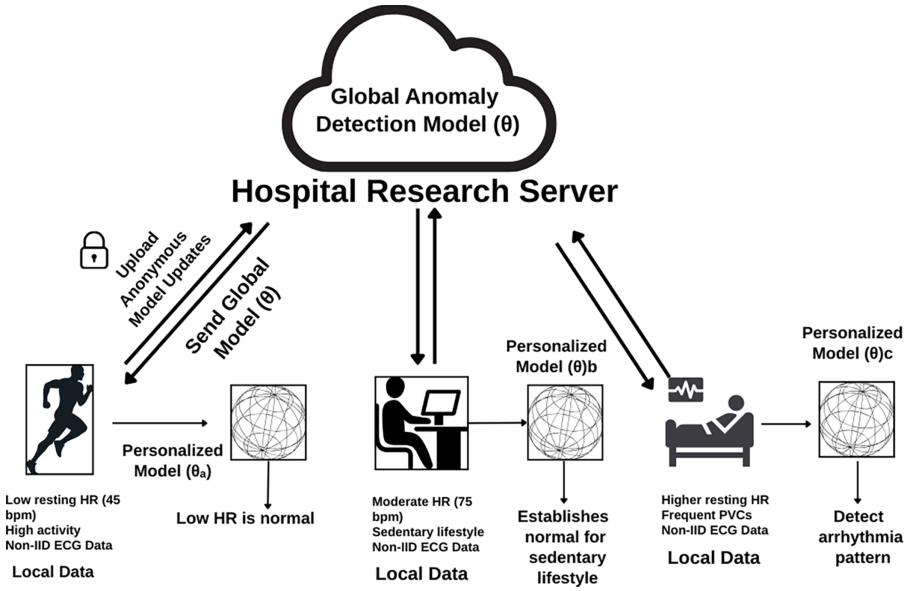

Figure 10: Personalized federated learning in healthcare

Table 3: Federated learning architectures with data distributions and associated security challenges

Architecture type	Data distribution	Main security/Privacy challenges	Example threats
Horizontal FL	Same features, different samples	Model update leakage, poisoning attacks	Inference attacks, back-door/poisoning
Vertical FL	Different features, same samples	Secure feature alignment, information leakage	Feature inference, collusion attacks
Federated Transfer Learning	Different features & samples	Cross-domain privacy, transfer attack surfaces	Model inversion, transfer attacks
Personalized FL	Heterogeneous data, personalized objectives	Client-specific leakage, limited generalization, personalization risks	Gradient leakage, personalized model inference

5.5 Synchronization Strategies: Synchronous vs. Asynchronous

5.5.1 Synchronous Federated Learning (SFL)

In Synchronous Federated Learning (SFL), clients train and send updates simultaneously. The central server aggregates updates only after receiving contributions from all predefined clients in that round. This ensures all updates are based on the same global model, leading to consistent learning. However, the system must wait for the slowest clients (stragglers), causing delays and wasting resources [31].

Fig. 11 illustrates Synchronous Federated Learning: clients of varying capabilities, including stragglers, upload their model updates simultaneously to a central server, which waits for all participants before proceeding with aggregation. This design ensures model consistency but may introduce delays due to slower clients.

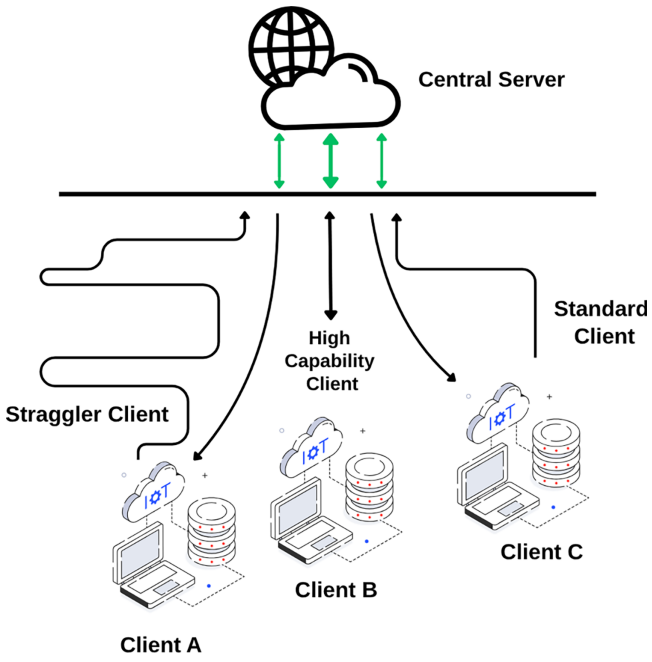

Figure 11: Synchronous federated learning

5.5.2 Asynchronous Federated Learning (AFL)

Asynchronous Federated Learning (AFL) allows clients to train independently and send updates when ready. The server updates the global model immediately upon receiving any update, without waiting for others. Clients download the current global model, train locally, and send updates back. The server immediately applies each update, allowing clients to participate freely based on availability.

Fig. 12 illustrates Asynchronous Federated Learning, highlighting the interaction between a central server and a diverse set of clients with varying statuses, including stable clients, unstable clients, and stragglers. The server processes model updates as they arrive, accommodating client heterogeneity and avoiding delays.

The core benefit of AFL is scalability, as fast clients do not wait for stragglers [74,75]. However, this introduces "model staleness," as clients may train on outdated global models, which can harm convergence and stability. AFL is useful in scenarios with unpredictable client availability, such as in large-scale IoT systems or with edge devices.

Table 4 presents a simple comparison of synchronous and asynchronous federated learning in terms of coordination needs, waiting time, update behavior, and practical suitability.

5.6 Privacy and Security in Federated Learning

5.6.1 Threat Models in Federated Learning

The security and privacy of FL systems are evaluated against two foundational adversary models, which define the goals of corresponding defenses:

- **Honest-but-Curious (or Semi-Honest):** This model assumes an adversary, typically the central server, that strictly adheres to the FL protocol's steps. However, it is "curious" and will leverage the information it legitimately receives (e.g., model updates) to passively infer sensitive information about the clients'

private data, such as attempting to reconstruct training samples. Defenses against this model focus on privacy.

- **Byzantine (or Malicious):** This adversary actively works to disrupt, corrupt, or control the learning process and does not follow the protocol. This adversary can be a client or the server, and may take arbitrary actions such as sending intentionally malformed data (data poisoning) or carefully crafted model updates (model poisoning) to degrade the global model's performance or insert a hidden backdoor.

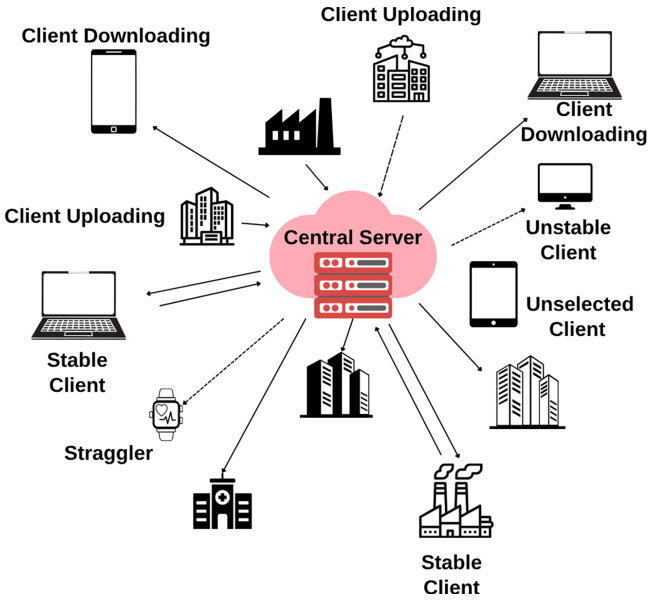

Figure 12: Asynchronous federated learning

Table 4: Comparison between synchronous and asynchronous federated learning

Feature	Synchronous FL	Asynchronous FL
Client coordination	High (all trains at once)	Low (clients act independently)
Waiting time	Can be high (waits for stragglers)	Low (no waiting)
Model updates	After receiving all updates	Immediately upon receiving an update
Update freshness	Always fresh	Can be stale
Use case fit	Controlled environments	Real-world, large-scale, dynamic systems

5.6.2 Data Leakage

Data Leakage refers to the unintentional leak of sensitive information from model updates, even when raw data is not shared. There are several ways for data leakage to happen:

Membership Inference Attacks (MIA): An adversary attempts to determine if a specific data sample was used in the training set, often by exploiting differences in model behavior on seen vs. unseen data [76].

Gradient Leakage (Model Update Leakage): Shared model updates can be exploited to reconstruct original training data (e.g., images or text) using techniques like Deep Leakage from Gradients (DLG) [77].

Fig. 13 illustrates an information leakage attack in an FL system. The server broadcasts the model (Step 1), and clients send back updates (Step 2). A malicious attacker intercepts these updates (Step 3) and analyzes them to infer or reconstruct sensitive information from the client's private data, leading to information leakage (Step 4).

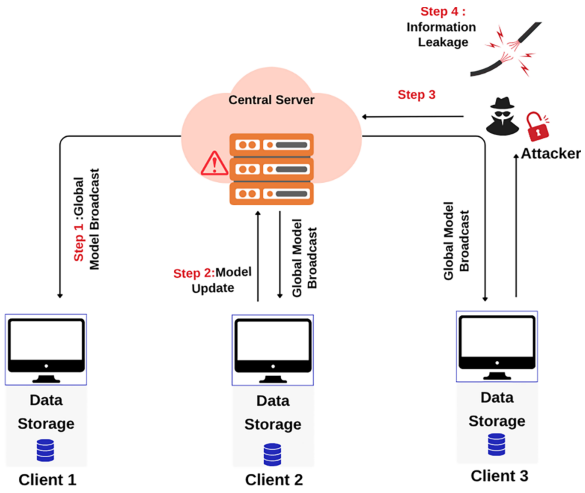

Figure 13: Data Leakage attack in federated learning

Property Inference Attacks: An adversary infers high-level statistical properties of the training data (e.g., demographic attributes) that may not be related to the model's primary task [13].

Reconstruction Attacks: An attacker attempts to recreate full or partial input samples from information leaked via model updates or predictions [78,79].

5.6.3 Poison Attack

A poison attack occurs when malicious clients intentionally manipulate the training process by sending harmful data or updates, aiming to degrade the global model's performance or introduce specific errors. This threat is significant in FL because the server cannot see the clients' raw data and must trust their updates.

Fig. 14 illustrates a poisoning attack: an adversarial client injects malicious updates (Step 1), which are then aggregated during the federated averaging process (Steps 2–3). These poisoned updates contaminate the global model (Step 4), affecting all honest clients.

Data Poisoning Attacks: Attackers poison their local training data (e.g., by flipping labels) to make the model learn incorrect patterns [80].

Model Poisoning Attack: The attacker crafts a malicious model update directly, designed to reduce accuracy or introduce a backdoor [81].

Mitigations include robust aggregation methods (e.g., Krum, trimmed mean), anomaly detection to flag suspicious updates [82], client reputation systems, and privacy mechanisms like differential privacy or secure aggregation [83].

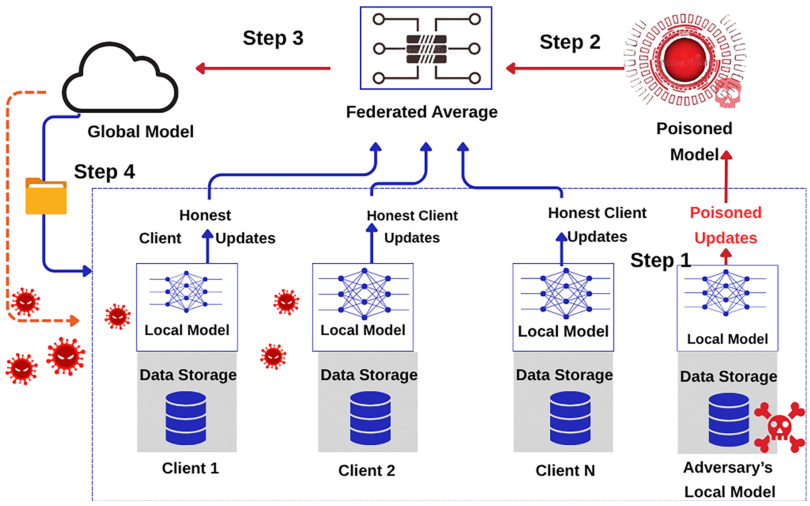

Figure 14: Poisoning attack in federated learning

6 Literature Reviews on Federated Learning Architectures: VFL, HFL, FTL, PFL

6.1 Vertical Federated Learning (VFL)

6.1.1 Architectural Advances in VFL with Heterogeneous Participants

Feng et al. in 2024 [84] propose MMVFL, a novel vertical federated learning (VFL) framework that supports multiple participants and multi-class classification tasks. This addresses the critical limitation of Traditional VFL methods, which are typically limited to two participants and binary classification, which restricts their applicability in real-world scenarios where data is distributed across many organizations and involves more complex classification problems. MMVFL addresses the challenge of enabling collaborative learning across decentralized data silos where participants share the same sample IDs but have different feature spaces while preserving privacy and extending scalability to multi-class classification tasks and multiple participants. The framework integrates multi-view learning (MVL) principles into the VFL setting, which enables label information to be shared from the label-holding participant to others via a privacy-preserving optimization process that learns pseudo-label matrices and applies sparsity constraints to transformation matrices. MMVFL methodology involves all K participants to train a model locally to generate pseudo-labels (Z_k). These locally predicted labels are then sent to a central server, which facilitates the creation of a global prediction, ensuring that raw data and true labels remain with their respective owners. The optimization process is defined by an objective function that minimizes classification errors along with a sparse regularization term to characterize feature importance. This is achieved by relaxing the constraints and employing an alternating optimization approach to iteratively update the local models (W_k), the local pseudo-labels (Z_k), and the global pseudo-label matrix (Z). MMVFL is presented as VFL framework capable of handling multi-class problems with multiple participants, and allows for easy integration with other communication and security techniques, with a feature selection scheme to quantify feature importance. However, its current implementation focuses on proof-of-concept with a simple linear feature selection method and does not yet incorporate advanced security protections or address potential communication bottlenecks caused by stragglers.

Wang et al. in 2025 [85] propose PraVFed, a framework for heterogeneous vertical federated learning (VFL) that addresses model heterogeneity and high communication costs. PraVFed allows each passive party to use its own unique, heterogeneous model and train it locally for several rounds. To reduce frequent

communication, the active party sends labels protected by differential privacy to the passive parties. After training, each passive party creates an embedding of its local data, which is then masked using a secure cryptographic technique to protect feature privacy. These masked embeddings, along with a weight that reflects how well each local model performs, are sent to the active party, which aggregates them using the provided weights. This aggregated information is then used to train the final global model. The framework achieves a 70.57% reduction in communication cost on the CINIC10 dataset. While it provides formal privacy guarantees for both feature and label information, the approach suffers from memory overhead from handling embedding values from all clients.

Zhang et al. in 2024 [86] introduce HeteroVFL, a VFL framework for two-level distribution in Industry 4.0, extending split learning with prototype aggregation and DP-protected smashed data sharing. Within regions, clients send smashed data; the server forms local class prototypes and aligns them to global prototypes via a joint loss combining classification and prototype consistency, enabling cross-region transfer without raw data or model sharing. On MNIST, Federated MNIST (FEMNIST), and CIFAR-10, HeteroVFL surpasses Split and SplitFed (e.g., 96%–97% vs. 90%–89% at n = 3) with lower communication and faster time-to-accuracy despite more epochs, and degrades less as regions grow. DP noise preserves privacy but reduces accuracy and slows convergence; $\epsilon \approx 2$ nearly matches non-DP, while small k (≤ 20) harms accuracy, exposing privacy-utility and data-quantity tradeoffs. Limitations include DP-induced precision loss for high-precision use, though communication remains lower than gradient aggregation baselines.

The architectural advances in VFL, such as MMVFL, PraVFed, and HeteroVFL, highlight the research trend in VFL, which is moving beyond the theoretical homogeneous two-party environment to the practical heterogeneous realities of multi-institutional collaboration. The reviewed papers approach heterogeneity from different angles. The MMVFL [84] framework expands VFL's scope to multi-party and multi-class scenarios by sharing pseudo-labels [84], which is a foundational step for wider applicability. PraVFed [85] and HeteroVFL [86] frameworks take this further by respectively addressing model and data distribution heterogeneity. These approaches are complementary rather than competing. For instance, PraVFed's focus on allowing diverse local model architectures could be combined with MMVFL's multi-party framework. HeteroVFL introduces a more complex, two-level hierarchical structure for Industry 4.0 that uses prototype-based aggregation for knowledge sharing across differing regions, which contrasts with the representation-learning approach in PraVFed. PraVFed and HeteroVFL integrate differential privacy to increase privacy and security, but the reliance on these abstractions introduces potential information loss, memory overhead, and degradation of model accuracy compared to fine-grained gradient updates.

6.1.2 Communication-Efficient and Privacy-Enhanced Protocols in VFL

Valdeira et al. in 2025 [87] introduces Error-Feedback Vertical Federated Learning (EF-VFL), a communication-efficient vertical federated learning (VFL) algorithm that employs error feedback-based compression to mitigate high communication overhead when training models across distributed, feature-partitioned clients. The method solves the inefficiency of existing VFL approaches, which either require vanishing compression errors for convergence or suffer from suboptimal convergence rates (e.g., $O(1/\sqrt{T})$) under practical bandwidth constraints. EF-VFL introduces an error feedback mechanism that updates surrogates of the transmitted representations to track and correct accumulated compression errors over training iterations, enabling nonvanishing compression while achieving a faster $O\left(\frac{1}{T}\right)$ convergence rate for smooth nonconvex objectives, which is an improvement over the $O(1/\sqrt{T})$ rate of previous compressed VFL methods. Under the Polyak-Łojasiewicz (PL) inequality, EF-VFL achieves linear convergence to the solution in the full-batch case, and more generally to a neighborhood proportional to the mini-batch variance. The method also supports private label scenarios, where only the server holds the labels, thereby broadening

its applicability. The paper provides theoretical guarantees, along with empirical validation on real-world datasets such as MNIST, CIFAR-100, and demonstrates that it is better than baseline frameworks like the standard VFL (SVFL) and compressed vertical FL (CVFL) in communication efficiency. However, potential sensitivity to highly aggressive compression may degrade performance and amplify distortion, especially under high compression error or noise.

Fan et al. in 2024 [88] propose FLSG, a defense strategy against passive label inference attacks in vertical federated learning, where existing privacy techniques like differential privacy fail because attacks are conducted entirely locally. FLSG intervenes at the server level by generating multiple random Gaussian gradients, computing cosine distances to the original gradients, and replacing the original with the most similar synthetic gradients when the distance falls below a threshold. Experimentation of the paper on six real-world datasets (CIFAR-10, CIFAR-100, CINIC-10, Yahoo Answers, Loan Default Prediction, BHI) shows that FLSG can reduce the success rate of passive label inference attacks by more than 40%, which outperformed other methods such as gradient clipping, noise gradients, and multistep gradients. Furthermore, it achieves this while using lower computational resources and having minimal impact on the model's accuracy. However, the study does not explore more complex situations where multiple malicious parties are involved.

Gong et al. in 2024 [89] propose a universal multi-modal vertical federated learning (VFL) framework based on homomorphic encryption to address the challenges of data silos, multi-modal data distribution, and privacy in collaborative machine learning. The solution couples a two-step transformer (cross-domain and multi-modal encoders) with a bivariate Taylor expansion that rewrites cross-entropy into addition/multiplication compatible with additively homomorphic encryption, enabling fully encrypted training without a third party. All exchanged gradients and losses are encrypted and masked under an honest-but-curious model. Experiments show accuracy gains of 1.33 on TWITTER-15 and 1.11 on TWITTER-17. On IEMOCAP, emotion accuracy rises by nearly 5, especially for unaligned data. While the methodology relies on approximating the cross-entropy function using a Taylor series expansion, the proposed protocol provides strong privacy guarantees, eliminating the need for a third-party coordinator for enhanced safety.

Valdeira et al. in 2025 [87], Gong et al. in 2024 [89], and Fan et al. in 2024 [88] highlight the two primary and technical challenges of communication overhead and privacy that are central to VFL's practical deployment. EF-VFL [87] mitigates the efficiency problem through a sophisticated error-feedback compression mechanism and achieves a faster theoretical convergence rate $O\left(\frac{1}{T}\right)$ than previous methods. In contrast, FLSG [88] and Gong et al.'s framework [89] prioritize privacy against distinct threats. While FLSG proposes a lightweight, server-side defense against passive label inference by substituting gradients, a direct countermeasure that requires minimal changes to the VFL protocol [88], Gong et al. aim for a more comprehensive, end-to-end privacy guarantee by using homomorphic encryption (HE) for a multi-modal VFL architecture. EF-VFL's compression is a complementary technique that could potentially be integrated with privacy protocols. However, the privacy methods themselves are largely competing. FLSG is computationally efficient and offers protection against a very specific type of attack, while Gong et al.'s framework provides maximal security at a higher computational cost, due to its use of Taylor series expansion to handle non-linear functions [89].

6.1.3 Generative Data Augmentation and Applied VFL in Specialized Domains

Xu et al. in 2024 [90] propose ELXGB, an efficient privacy-preserving XGBoost for VFL that secures alignment, training, and inference via a trusted authority for keys, Cloud Service Provider (Conditional Generative Adversarial Network (CGAN))-assisted PSI, and Paillier-plus DP-based node splitting under an honest-but-curious, non-colluding model. ELXGB builds the first tree with homomorphic encryption-based node split (HENS) and subsequent trees with differential privacy-baesd node split (DPNS), uses a gradient

cache to keep leaf weights lossless, and applies attribute and direction obfuscation to hide node attributes and model structure during inference. On Credit Card (30,000 samples, 23 features) and Bank Marketing (45,211 instances, 17 features), accuracy nearly matches vanilla XGBoost (≈83% and ≈90% with the global proposal), while competing schemes incur heavy training time (e.g., T = 5, D = 3 on Credit Card: primitive XGBoost ≈ 1 s vs. PIVODL ≈ 480 s and SecureBoost ≈ 209 s), and ELXGB's communication remains stable and about five times lower than SecureBoost. The model achieves high efficiency by limiting computationally intensive encryption to only the first tree and by enabling a centralized inference process. However, the framework's security model does not account for active attacks (such as poisoning or backdoors) and relies on an assumption of non-collusion between parties.

Xiao et al. in 2025 [91] address insufficient overlapping data in VFL by proposing FeCGAN, which combines a central generator with distributed discriminators, FedKL aggregation that weights clients by KL divergence, and VFeDA to synthesize pseudo features for non-overlapping segments, thereby reducing learning divergence without exposing raw data. On Fashion-MNIST and CIFAR-10, FeCGAN raises classification accuracy by about 14.9%–39.2% over centralized/distributed Generative Adversarial Network (GAN) baselines, outperforms Local-GAN and federated learning with zero-shot data augmentation at the clients (Fed-ZDAC) [92] by up to 10.45%, and achieves higher Inception Scores and lower Fréchet Inception Distance (FID), while sparse representations converge faster (≈20 epochs), improve accuracy by ≈18% over Principal Component Analysis (PCA), and cut FID by ≈25%. Gains persist under IID and non-IID splits, but augmentation weakens for participants whose distributions are far from the global mixture, and further optimization is needed to curb communication overhead.

Yan et al. in 2024 [93] propose Fed-CRFD, a cross-modal VFL framework for collaborative Magnetic Resonance Imaging (MRI) reconstruction across hospitals with heterogeneous modalities and limited multimodal overlap, addressing domain shift that standard FL fails to resolve. Fed-CRFD disentangles modality-specific and modality-invariant features via an auxiliary modality classification and enforces cross-client latent consistency on overlapping patients to align invariant representations during aggregation. On fastMRI (T1w/T2w) and a private clinical dataset, it outperforms FL baselines, raising Peak Signal-to-Noise Ratio (PSNR) from 33.30 to 35.35 dB over FedAvg on fastMRI and from 29.95 dB to 30.95 dB on the private set ($p < 0.05$), with Similarity Index Measure (SSIM) gains and robustness when vertical overlap drops from 10% to 2%. With three clients, PSNR improves from 33.77 to 35.65 dB and Structural SSIM from 0.9192 to 0.9389, and downstream tumor segmentation achieves higher Dice and lower Average Symmetric Surface Distance (ASSD). Overheads are modest (≈1.1 M latent-feature bytes per round on fastMRI; slight training-time increase) but rely on a trusted server for latent feature handling, and applicability may diminish when all clients share one modality.

Xu et al. (ELXGB) [90] and Xiao et al. (FeCGAN) [91] show a shift from creating general-purpose frameworks to developing highly specialized solutions for specific algorithmic, data, and domain challenges. ELXGB provides a highly secure and efficient protocol specifically for training XGBoost models, a widely used model type where secure non-linear operations are a major hurdle [90]. FeCGAN tackles the fundamental problem of insufficient sample overlap by using a distributed generative model to augment the client datasets themselves [91]. These two approaches are complementary, and a system could potentially use FeCGAN for data augmentation before applying a secure training protocol like ELXGB. Yan et al.'s Fed-CRFD [93] showcases a complex domain challenge in medical imaging by redesigning the model architecture to include feature disentanglement, creating a shared representation space from cross-modal data [93]. While all three aim to enhance VFL's practicality, their approaches are fundamentally different: ELXGB customizes the process, FeCGAN enhances the data, and Fed-CRFD re-engineers the model. Together, these papers illustrate a shift towards building VFL frameworks that will likely focus less on foundational protocols and

more on their sophisticated integration with specific machine learning paradigms and the unique constraints of high-impact application domains.

6.1.4 Assessing Privacy-Communication-Utility Trade-Offs in VFL Deployment

Assessment of privacy-communication-utility trade-offs in real-world VFL deployments requires practitioners to evaluate three competing metrics simultaneously and their interdependencies. The reviewed VFL protocols illustrate a framework for such assessment: first, quantify computational cost by measuring per-round client-side overhead (seconds or CPU cycles) relative to baseline FedAvg, quantify communication cost as bandwidth consumed per round (kilobytes or reduction percentage), and quantify utility loss as accuracy or convergence degradation. Second, practitioners must recognize that trade-off severity is data- and model-dependent, not universal. For instance, HeteroVFL demonstrates that DP integration with ($\epsilon \approx 2$) achieves near-baseline accuracy (96%–97% on MNIST/FEMNIST/CIFAR-10 vs. 90%–89% for unprotected methods), yet small participant counts (($k \leq 20$)) amplify accuracy harm disproportionately, exposing a privacy-utility-data-quantity interaction [86]. Similarly, EF-VFL's error-feedback compression achieves $O\left(\frac{1}{T}\right)$ convergence—faster than prior $O\left(\frac{1}{\sqrt{T}}\right)$ methods—but exhibits sensitivity to aggressive compression ratios, requiring empirical tuning per deployment rather than universal defaults [87]. Third, threat model specificity determines which trade-off is most relevant: FLSG reduces passive label inference attacks by over 40% with minimal overhead, but provides no defense against active multi-party collusion attacks [88], rendering its computational efficiency meaningless in high-threat environments. In contrast, Gong et al.'s homomorphic encryption framework provides end-to-end privacy guarantees without third-party coordinators, but introduces per-round latency measured in minutes from Taylor series approximation of non-linear functions, making practical real-time deployment infeasible for most applications [89].

Practitioners deploying VFL should first define deployment constraints—acceptable accuracy loss threshold (typically 1%–5%), bandwidth budget (megabytes per round), device computational budget (seconds per round), and privacy requirements (threat model and sensitivity level)—then match these to frameworks optimized for those constraints. PraVFed's 70.57% communication reduction is valuable only when bandwidth is the bottleneck; its memory overhead from embedding aggregation becomes prohibitive in cross-organizational settings with dozens of passive parties [85]. The assessment process should prioritize compression and selective privacy for non-sensitive model components rather than uniform heavyweight protection. Research on adaptive mechanisms Section 8.1.2 shows that risk-aware, dynamic privacy allocation achieves better privacy-utility balance than fixed-budget approaches across all training rounds [94,95]. Finally, evaluation must include realistic non-IID data splits and multi-round aggregation, as trade-offs under synthetic homogeneous splits diverge substantially from production deployments where client data distributions vary by order of magnitude and model degradation compounds across rounds.

6.2 Horizontal Federated Learning (HFL)

6.2.1 Architectural Advances in Hybrid and Heterogeneous Federated Learning

Yu et al. in 2025 [96] propose a communication-efficient hybrid federated learning framework designed for e-health applications where data exhibits a complex three-tier horizontal-vertical-horizontal distribution structure. The issue addressed is that existing Horizontal or Vertical Federated Learning (HFL/VFL) methods are inefficient for horizontal-vertical-horizontal structure, and either demand excessive communication of raw data for accuracy or suffer from poor model performance. The authors solve this by introducing a framework that has two aggregation phases and one intermediate result exchange. The vertically partitioned data between hospitals and patient-owned wearable devices is handled by exchanging intermediate results to

train sub-models collaboratively without sharing raw data. A local aggregation phase at an edge node creates a unified device-side model to manage the first layer of horizontal partitioning across wearable devices within a single hospital's purview to enhance training efficiency. Finally, a global aggregation phase handles the second horizontal partitioning layer across different hospital-patient groups to create a generalized model. This process is operationalized through a newly developed Hybrid Stochastic Gradient Descent (HSGD) algorithm. The paper provides a convergence analysis of the HSGD algorithm and uses the results to come up with three adaptive strategies for tuning crucial hyperparameters (aggregation intervals P and Q, and learning rate) to balance communication cost and model accuracy. However, the theoretical convergence proof is provided for the i.i.d., whereas experiments are run on non-IID data, the theoretical guarantees may not fully extend to these cases.

Peng et al. in 2024 [97] propose a Hybrid Federated Learning algorithm for Multimodal Internet of Things (IoT) Systems Hybrid Federated Learning Model (HFM) to address the challenge of efficiently training models on data that is distributed across both different devices (sample space) and different data types or sensors (feature space) under stringent resource constraints. The core problem is that the trade-off between the high computational demands of processing high-dimensional multimodal data such as images, audio, and sensor signals, and the limited memory and compute capabilities of IoT devices and edge servers along with complex data distribution, where data is partitioned both horizontally (across different locations or silos, such as households or factories) and vertically (across different sensor modalities, like cameras and microphones within a single silo), and made more difficult by the presence of non-independent and identically distributed (non-IID) data across silos. HFM solves this by creating a two-tiered federated system where it first applies Vertical Federated Learning (VFL) to train models on different data modalities, such as image, audio from different IoT devices without sharing raw data and distributing the computational load across the feature space (different modalities) within each silo such as household or factory, allowing IoT devices to process their own data modalities and share only embeddings. It then employs Horizontal Federated Learning (HFL) to aggregate the learned models from all the different 'silos' (e.g., multiple households) to build a single global model. VFL partitions computing resources across feature spaces (modalities) among IoT devices within each silo, while HFL distributes learning across sample spaces (silos) via a global server. The paper provides a theoretical analysis, proving that the convergence of HFM depends on the frequency of VFL and HFL communications as well as the number of vertical and horizontal partitions, and it also empirically demonstrates that HFM outperforms baseline methods in terms of convergence rate and error on two public multimodal datasets. However, the current theoretical framework and experiments assume periodic synchronous updates and do not address potential real-world issues arising from heterogeneous IoT devices or modalities.

Yi et al. in 2023 [98] propose FedGH (Federated Global prediction Header), a framework for heterogeneous federated learning (FL) that addresses the challenge that traditional horizontal FL methods often require the server and clients to use the same model structure, which is impractical due to varying device capabilities. To solve this, FedGH allows each client to use a client-specific heterogeneous feature extractor while sharing a homogeneous global prediction header. During training, each client trains its local model and computes a local averaged representation (LAR) for each data class it possesses and uploads these to the server along with corresponding labels. The server aggregates LARs across clients to train a generalized global prediction header via gradient descent. This globally trained header, which encapsulates knowledge from all participating clients, is then distributed back to the clients, where the global header substitutes the local headers. This approach significantly reduces communication and computation cost by avoiding the transfer of large model parameters and instead sending only the small average representations. This method also strengthens privacy by sending only the abstract representation instead of more revealing data

or model details, and does not depend on the availability of a public dataset as it trains the global prediction header using class-wise averaged representations from clients' local data instead of requiring public data. The FedGH approach distinguishes feature extraction from prediction, which enables model heterogeneity and allows knowledge transfer through the shared header. Experimental results of FedGH show higher accuracy over state-of-the-art personalized FL methods, such as Federated Prototype Learning (FedProto), LG-FedAvg, on non-IID datasets, with up to 85.53% reduction in communication overhead. However, the authors acknowledge that creating a local averaged representation for a class with numerous data samples might lead to information distortion.

The advancements in HFL highlight a trend of dismantling rigid architectural assumptions in HFL to accommodate the profound heterogeneity of real-world applications. The works by Yu et al. [96] and Peng et al. [97] move beyond the pure HFL/VFL dichotomy by treating HFL and VFL not as monolithic options, but as modular components to be assembled based on the problem's structure. Yu et al. design a specific three-tier H-V-H structure for e-health data [96], while Peng et al. propose a two-tiered VFL-within-HFL architecture for multi-modal IoT systems [97]. These frameworks highlight that as FL is applied to more complex scenarios, rigid distinctions between HFL and VFL are dissolving in favor of flexible, multi-layered designs. In contrast, FedGH [98] tackles heterogeneity at the model level by decoupling the model into a client-specific feature extractor and a shared global prediction header, which allows clients with varying resources to participate effectively using a highly communication-efficient knowledge transfer mechanism. These methodologies are potentially complementary; for instance, the HFL aggregation stage in the hybrid frameworks proposed by Yu et al. or Peng et al. could leverage a FedGH-like mechanism to allow participating silos to use different model architectures. Hybrid models offer tailored solutions for complex data distributions but increase architectural complexity, while FedGH elegantly solves model heterogeneity at the risk of information loss via its class-averaging mechanism.

6.2.2 Communication Optimization and Feature Selection in HFL

Banerjee et al. in 2024 [99] propose Fed-FiS and Fed-MOFS, two cost-efficient feature selection (FS) methods for horizontal federated learning (HFL), where clients share a common feature space but possess different data samples to address the challenge of identifying a relevant and non-redundant subset of features that can be commonly used across all clients to enhance global model performance to reduce training time and computational costs. The authors also point out that existing FS methods are designed for centralized systems, which fail to account for data heterogeneity and communication constraints in HFL systems where clients share a common feature space but suffer from statistical divergence and local feature selection bias. Fed-FiS utilizes mutual information (MI) to quantify feature relevance locally, and clustering for local feature selection at each client, which is followed by a global ranking based on aggregated feature importance scores. Fed-MOFS extends this with multi-objective optimization (Pareto optimization) to explicitly maximize feature relevance, minimize redundancy, and client-specific variations during global selection, which results in a globally ranked feature set derived from Pareto fronts. The paper evaluates both models on diverse datasets such as NSL-KDD99, IoT, and credit scoring under IID/non-IID distributions, and demonstrates efficiency by achieving a 50% reduction in feature space and twice the speedup of comparable methods with improved accuracy and convergence. However, the dependency on mutual information measures may introduce computational overhead on client devices.

Zhang et al. in 2023 [43] propose FSHFL, an unsupervised federated feature selection framework for horizontal federated learning (HFL) within Internet of Things (IoT) networks that addresses the challenges of high-dimensional, non-IID, and unlabeled data where all features in a distributed client dataset do not contribute positively to the global model while some degrade model performance, increase training time,

and energy consumption. The paper solves this by using a feature average relevance one-class support vector machine (FAR-OCSVM) algorithm that detects and removes irrelevant or anomalous features based on their average correlation with other features. Subsequently, a feature relevance hierarchical clustering (FRHC) algorithm groups the correlated features to highlight representative clusters that emphasize representative features that capture the data's intrinsic structure. The most informative feature clusters from each client are then intersected at the server to identify globally relevant federated features, which are sent back to clients for local model training without sharing raw data or labels. Experimental results on four IoT datasets (Bot-IoT, ACC, KDD99, DEFT) demonstrate that FSHFL improves global model accuracy by up to 1.68%, reduces training time by 6.9%, and lowers energy consumption by 2.85% compared to Fed-Avg and 68.39% compared to Fed-SGD. While the method works well on non-IID and unlabeled data and enhances privacy, explicit fairness among clients is not considered, and may not generalize across all IoT domains.

Dai et al. in 2023 [100] introduce Federated cross-domain sequential recommendation (FedCSR), a clustered sampling framework based on a rotation mechanism that addresses the challenges of data heterogeneity and client representativity in Horizontal Federated Learning (HFL). The paper solves the problem of unstable convergence and reduced accuracy of existing clustering and sampling methods in HFL caused by non-IID data distribution. In the FedCSR approach, each client uploads Gaussian Mixture Model (GMM) parameters that represent its data distribution to the server before training. The server uses these parameters to calculate the KL Divergence between clients to quantify how different their data distributions are, which enables dynamic clustering that adaptively adjusts the number of clusters and intra-cluster sampling probabilities. Clients are further grouped by their cosine similarity of model updates, and a rotation-based sampling mechanism balances exploration and exploitation by selecting clients based on distribution difference while maintaining fixed sampling sizes. This ensures that clusters are adaptively and representatively sampled across communication rounds, mitigating biases from static clustering and enhancing model generalization. Experimental results on MNIST and CIFAR10 datasets show that FedCSR outperforms baseline methods such as FedAvg, FedProx, and CFL-L2 in convergence speed, stability, and accuracy under high data heterogeneity. However, the framework treats all clients in a group equally, which may limit the ability to adjust the aggregation coefficient for clients within the same group.

Zhang et al. in 2024 [101] propose a privacy-preserving data selection framework for both horizontal and vertical federated learning (HFL and VFL) scenarios to improve the convergence speed and accuracy of FL models by selecting high-quality training samples and features for a given FL task under a monetary budget. Poor-quality data, such as samples with incorrect labels, irrelevant features, or low content diversity, impacts badly on the performance of federated learning models. The authors address the problem of how to efficiently select relevant clients and their high-quality samples or features without compromising data privacy, especially in scenarios where clients' local data and the server's label set must maintain confidentiality. The paper presents a holistic approach by including a hierarchical client and sample selection mechanism based on statistical homogeneity and content diversity, using a private set intersection (PSI) protocols to identify clients and samples relevant to the target task without revealing private information along with a determinantal point process (DPP) based algorithm to select clients that maximize both statistical homogeneity (data distribution similarity) and content diversity, under a budget constraint in HFL settings. For VFL, the algorithm uses PSI to match sample IDs and then applies a Gini impurity-based method (with homomorphic encryption) to select important features securely, while minimizing the inclusion of noisy or irrelevant features. During training, the system dynamically selects important clients and samples based on their estimated impact on the global model using gradient by calculating an importance score for each sample which is defined as the upper bound of the gradient norm of the loss with respect to the pre-activation outputs of the model's last layer, while filtering out erroneous samples that tend to exhibit abnormally large

importance values. The experimental validation on a real-world Artificial Intelligence of Things (AIoT) system with 50 clients demonstrates significant improvements in model accuracy, convergence speed, and reduced computation/communication costs compared to baseline methods. However, the effectiveness of the erroneous sample filtering relies on setting a proper threshold for importance values, which may require careful tuning based on the dataset and model.

Banerjee et al. [99] and Zhang et al. [43] focus on feature selection, while Dai et al. [100] and Zhang et al. [101] concentrate on client and data point selection. While both aim to reduce dimensionality, the feature selection methods differ in their technique. Banerjee et al. propose a generalized multi-objective optimization approach, whereas Zhang et al. [43] propose an unsupervised method that is specifically optimized for IoT data. These methodologies are complementary. A trailblazing HFL pipeline could first employ a feature selection method like Fed-FiS to create a common, efficient feature space. Subsequently, a client selection strategy like FedCSR could be used to choose the most representative clients for a given round. Finally, the sample-level filtering proposed by Zhang et al. [101] could be applied by the selected clients during their local training. This integrated approach that combines feature, client, and data selection represents a possible direction for building HFL systems that are communication-efficient while being resilient to the challenges of severe data heterogeneity.

6.2.3 Application-Oriented Frameworks and Security Enhancement

Qiu et al. in 2024 [102] propose a federated learning (FL) framework to facilitate multi-institutional collaboration for heart sound abnormality detection, aiming to address early cardiovascular disease diagnosis using heart sound data, with a focus on the issues of data privacy and the interpretability of AI models. The authors address the problem of "data islands," where valuable medical data is stored within individual institutions due to privacy regulations, which limit the AI model improvement, as AI models require a large and diverse dataset to predict accurately. To solve the difficulty in training effective AI models for cardiovascular disease diagnosis due to data scarcity, non-independent and identically distributed (non-IID) data, the authors implement both horizontal FL (HFL) and vertical FL (VFL) strategies using a SecureBoost-based ensemble learning approach that enables collaborative model training without direct data exchange. For HFL, where institutions have different patients but similar data features, a privacy-preserving feature ID-based hash map and Rivest–Shamir–Adleman (RSA) encryption-based aggregation method that aligns feature spaces across participants without information leakage is used. For VFL, where institutions hold different types of data for the same patients, the framework enables the integration of comprehensive features as well as tackles the "black-box" nature of AI models through application of Shapley values (SHAP) that preserves privacy by calculating feature importance on binned data from host parties, balancing model interpretability with data security and ensuring transparency in clinical decision-making. The work shows its application to real-world, multi-center heart sound databases, with its practical privacy-preserving strategies and real-world data validation. However, the HFL experiments involve only five institutions, which may not fully represent larger-scale deployments.

Pang et al. in 2024 [103] propose a horizontal federated learning (HFL) privacy-preserving framework integrated with an efficient fully convolutional network (e-FCN) that addresses the issue of privacy preservation for crowd counting in smart surveillance systems. The paper aims to solve the problem of privacy risks of collecting and centrally training crowd counting models on surveillance data, which can often contain personal information such as faces, clothing, body posture, and environmental details. Traditional crowd counting methods either do not fully resolve privacy concerns or require data transmission that risks data leakage, along with the challenge of non-independent and identically distributed (non-IID) data across different surveillance devices, which makes it complicated for crowd counting tasks using federated

learning. The proposed framework allows multiple surveillance clients to collaboratively train a global crowd counting model without sharing raw video data, where only model parameters are shared between clients and a central server, which aggregates them to update the global model. The e-FCN is designed to be a lightweight encoder–decoder–based architecture with less parameters to make it usable for client devices that have limited resources. It uses a truncated VGG-16 as the encoder and deconvolution layers as the decoder to generate density maps for crowd counting. To evaluate the framework under realistic non-IID conditions, four non-IID partitioning strategies, feature-skew, quantity-skew, scene-skew, and time-skew, are introduced to simulate heterogeneous data distributions across seven real-world surveillance datasets. The framework is evaluated using four FL algorithms, such as FedAvg, FedProx, FedNova, and SCAFFOLD, under the non-IID settings, which provides a standardized benchmark for federated crowd counting. However, relying only on model parameter aggregation may not fully mitigate the risks of gradient-based inference attacks.

Noman et al. in 2023 [104] propose a federated learning-based approach leveraging blockchain technology to address the challenges of building an accurate global classification model for multi-class respiratory diseases, such as pneumonia, tuberculosis, and COVID-19, in the healthcare sector. The primary problem is the scarcity and diversity of medical data due to privacy concerns and legal obstacles that limit data sharing among institutions. Although federated learning is a promising solution for distributed model training while preserving privacy, an effective aggregation process for multi-class and heterogeneous medical data remains a challenge. The authors tackle this challenge by introducing a new "weight manipulation technique" that combines both the test accuracy and the amount of data from each local model when merging them, instead of just looking at data quantity. They use a deep neural network (DNN) for classification and test their approach on a balanced dataset of 5000 chest X-ray images, with 1000 images in each of the five categories: COVID-19, Pneumonia, Tuberculosis, Lung Opacity, and Uninfected. Instead of sharing any raw patient data, the system uses blockchain to help institutions trust each other by saving secure, unchangeable summaries (called cryptographic hashes) of the model's weights and related details on a digital ledger. To further encourage participation, an adaptive incentive mechanism is introduced, rewarding contributors based on their model's performance on a global test set and the amount of data contributed, with extra emphasis on recall, precision, and F1-score. The federated model with the proposed weight manipulation technique achieved a testing accuracy of 88.10% for five classes, closely matching the 88.60% accuracy of a centralized model trained on all data. The proposed weight manipulation technique enhances the aggregated model's performance on non-IID data, outperforming the standard FedAvg algorithm in multi-class classification scenarios. However, the dataset used is actually made by combining publicly available sources, which may not fully reflect the real-world challenges and data differences that would come from working with truly independent institutions.

Kandil et al. in 2024 [105] propose an IoT-enabled Vendor Managed Inventory (VMI)-based framework leveraging Horizontal Federated Learning (HFL) to improve data sharing in supply chain systems by addressing the privacy and scalability issues, particularly those inherent to blockchain-based VMI systems. The authors identify the issues with blockchain, such as poor scalability, high energy consumption, data privacy risks, and regulatory complexities, which pose barriers to efficient, real-time inventory management in large-scale supply chains. To solve this, the paper introduces a multi-layered network architecture that utilizes Horizontal Federated Learning (HFL) that integrates IoT sensors, edge computing, and cloud-based federated learning. This multi-layered network architecture consists of a physical layer with IoT sensors such as RFID tags, shelf sensors, vehicle sensors for real-time inventory and environmental data collection, an edge computing layer for local data processing with model training at each entity for quick response and privacy, a core network which acts as a bridge for data transfer between local and global models, a cloud computing layer for Aggregating features from all entities to train a global model for demand forecasting and

inventory optimization, and a VMI interface for user interaction. A detailed comparative analysis highlights that this HFL's superiority over blockchain in terms of data privacy, scalability, cost-efficiency, and lower energy consumption for VMI applications. However, the adoption of HFL faces challenges such as the initial costs of new technology, a lack of standardization, and potential security vulnerabilities in the federated learning process.

Qiu et al. in 2024 [102] and Noman et al. in 2023 [104] both tackle healthcare challenges by applying different architectural enhancements. Qiu et al. focus on interpretability through privacy-preserving SHAP values, while Noman et al. integrate blockchain to enhance security and trust, along with an incentivized weight manipulation technique to handle non-IID medical data better. These approaches are not mutually exclusive. In fact, a system could leverage blockchain for security while incorporating interpretability methods. Pang et al. and Kandil et al. in 2024 [105] highlight HFL's role in the IoT ecosystem. Pang et al. (2024) [103] provides a benchmark for federated crowd counting that demonstrates HFL's utility in privacy-sensitive video surveillance. Kandil et al. present HFL as a direct, more efficient alternative to blockchain for data sharing in supply chains, a notable contrast to Noman et al.'s use of blockchain as a complementary technology.

6.3 Federated Transfer Learning (FTL)

6.3.1 Architectural Advances in Federated Transfer Learning

He et al. (2023) [106] propose FT-FLC, an improved algorithm for image classification that is based on Federated Transfer Learning (FTL). The main challenge is tackled by the authors, enabling effective machine learning when data is distributed across multiple entities and cannot be merged due to privacy or legal constraints. To address this, the authors propose the FTL-FLC algorithm, which combines federated learning and transfer learning. FTL-FLC establishes a system consisting of several clients, a central server, and a distinct origin model. Each client trains a local model using its own "source dataset" (CIFAR-10 and CINIC-10) and transmits only the model weights, not the raw data, to the central server. This server connects these weights using the FedAvg algorithm and updates the Origin model accordingly. This updated origin model then fine-tunes these combined weights on a smaller, labeled "target dataset" (STL-10) before sending the refined parameters back to the server. After which, they are sent back to all clients for the following training iteration. A notable part of the approach is a pre-training step using an encoder-decoder architecture on the unlabeled portion of the STL-10 dataset, which helps establish a strong initial model for all participants. The method's main advantage is its ability to boost classification accuracy on the target dataset by 8.17%, reaching a top accuracy of 83.54%, while also speeding up convergence and maintaining solid performance on the source datasets. That said, the paper deliberately avoids delving into detailed privacy-preserving techniques like homomorphic encryption, choosing instead to focus on presenting and evaluating the algorithm itself.

Naseh et al. (2024) [107] propose a practical implementation and relative performance analysis of distributed learning frameworks, specifically federated learning (FL) and federated transfer learning (FTL), designed for 6G Internet of Things (IoT) applications. The authors address the challenge of efficiently training machine learning models across diverse IoT devices, where traditional centralized learning is impractical because of high communication costs and network traffic, even standard FL faces challenges and struggles with slow learning and differences in client abilities. To solve this, they deploy FTL on a real-world testbed of heterogeneous low-power devices (Raspberry Pi, Odroid) and virtual machines, using pre-trained models to initialize the global parameters, which results in faster convergence, higher accuracy, and lower resource consumption compared to FL. The authors detail their methodology by training a Deep Neural Network (ResNet) on the CIFAR-10 dataset for an image classification task. To better reflect real-world conditions, they distribute the data unevenly among clients. In tests involving five clients, they show that FTL cuts

training time by about three times, reaching 83% accuracy compared to 60% with traditional FL, while also greatly reducing computational demands, memory use, temperature, power, and energy consumption. The main idea of this approach is to use federated transfer learning (FTL), where clients start with a pre-trained global model and fine-tune it using their own local data. This method is both more efficient and effective than traditional federated learning. because it requires training models from scratch and requires much more computing power. One of the strong points of the paper is its focus on real-world testing. Instead of just reporting accuracy and loss, the authors also look at practical factors like CPU load, memory use, temperature, power, and energy consumption. To test FTL's flexibility, they even ran an experiment with only three clients to simulate a mobile scenario where users drop off during training. In that scenario, FTL's accuracy dropped only slightly, from 83% to 81%, whereas standard FL saw a larger drop, going from 60% to 55%. This indicates that FTL effectively handles small-scale, low-participation settings better than traditional methods. These experiments were conducted on only a handful of client devices. It has been a concern how well the method would scale in larger deployments. The authors also mention that the effect of memory swapping on overall performance still needs to be explored further.

Rajesh et al. (2023) [108] introduce a Federated Transfer Learning (FTL) framework designed for network intrusion detection in Industrial Internet of Things (IIoT) environments. It addresses the key challenge of building effective machine learning-based intrusion detection systems in the face of limited and privacy-sensitive IIoT data, as well as the complexity of dealing with heterogeneous data sources. The proposed approach uses a client-server architecture and features a novel combinational neural network (Combo-NN) that integrates a ResNet-50 convolutional neural network (CNN) with a deep neural network (DNN) as its base model. In the training phase, data is split between the server and various clients. The server initially trains a global model on its own data and then distributes it to the clients for local fine-tuning. A key strength of the paper is its high-performance results on an IIoT dataset, where the FTL approach outperforms contemporary machine learning algorithms like Logistic Regression, Gaussian Naïve Bayes, and Random Forest. The authors claim this is the first application of FTL for IIoT network intrusion detection. However, the performance is slightly decreased across iterations due to class imbalance and the need for further validation on additional datasets against adversarial attacks.

Shahnazeer and Sureshkumar (2025) [109] propose a novel predictive healthcare framework for the early detection of multi-organ failure (MOF) by utilizing Federated Transfer Learning (FTL). The central problem addressed is the critical need for timely MOF identification in intensive care settings, which is often hindered by patient data privacy concerns and the difficulty of creating generalizable models from isolated datasets. The authors tackle this problem through a multi-step process that first collects and combines decentralized patient data, including vital signs, lab results, and demographics, from various healthcare institutions. Federated learning is used to protect privacy, which allows a global model to be trained on local data without the need to transfer raw data outside each healthcare institution and only model updates are shared centrally. The proposed framework enhances prediction accuracy by using transfer learning, which takes advantage of pre-trained models from similar medical areas to better understand MOF-specific data. Firstly, after cleaning and normalizing local data, a global model is built by combining updates from different local models. The updated local model provides real-time prediction and alerts clinicians when a patient's risk of MOF beyond a set threshold. This strategy not only supports data privacy but also ensures scalability and strength. However, more future work is needed to minimize false positives and further increase precision in MOF detection. And also, it does not compare its results in detail with other models.

The key benefits of using transfer learning in a federated environment are accelerated convergence speed and improved performance of machine learning models, especially in resource-constrained devices. A key trend is the use of pre-trained models to overcome the "cold start" problem in FL, where training models from

scratch is inefficient. He et al. (2023) [106] and Naseh et al. (2024) [107] both demonstrate this principle but in different contexts. He et al. use a source-target setup for image classification to improve accuracy on a distinct target dataset, while Naseh et al. provide a real-world implementation on heterogeneous IoT hardware, focusing on the practical improvements in training time, power consumption, and resource utilization over standard FL. Rajesh et al. (2023) [108] and Shahnazeer and Sureshkumar (2025) [109] show the application of this FTL to specific and critical domains such as network intrusion detection in IIoT and multi-organ failure prediction in healthcare, respectively. These application-driven papers are not proposing fundamentally new FTL architectures but are rather validating the effectiveness of the established FTL concept for solving pressing real-world problems in situations where data is both scarce and private.

6.3.2 Privacy Preservation, Security Mechanisms, and Robustness in FTL

Zhao et al. (2025) [110] propose a personalized label inference attack, named the Contrastive Meta Learning (CML) attack, to evaluate and exploit privacy risks in Federated Transfer Learning (FTL), a domain where such vulnerabilities have been underexplored. The authors identify the problem that parameter decoupling strategies in FTL, while effective for model personalization, can leak private label information through the fine-tuning process and gradient updates. The CML attack combines two main components: meta-classifiers for high-dimensional feature extraction with contrastive learning to disentangle personalized data representations from shared global patterns. Specifically, the attack leverages shadow datasets and differences in posterior output from the server's global model and clients' fine-tuned models to train a meta-classifier. This is further enhanced by contrastive learning, which is uniquely used to distinguish personalized information from global information by amplifying the differences between a client's local model and the global model. Evaluated on CIFAR-10/100 under non-IID data distributions, the CML attack achieves high success rates (79.11% on CIFAR-10 with $\alpha = 0.1$), outperforming baseline methods. The primary strengths of this work lie in its pioneering focus on FTL-specific privacy risks, introducing contrastive learning as a tool for on access to shadow data and the assumption of server-side knowledge of client model differences, which may not hold in realistic adversarial scenarios. A limitation of the paper is that while it successfully exposes a significant privacy threat, the evaluation of mitigation techniques against this potent attack is left for future work.

Wang et al. (2024) [111] propose a novel blockchain-enabled federated transfer learning (FTL) framework for anomaly (fault) detection in power lines. Dealing with the challenges of privacy, security, and model performance in centralized machine learning approaches. Here, the primary issue is detecting faults, especially partial discharges in covered conductors. Traditional centralized training of deep learning models on edge devices is constrained by computational limitations and raises significant concerns regarding data privacy and security. To overcome these issues, the authors introduce a novel decentralized approach that combines federated learning (FL) with transfer learning (TL), enabling edge devices to collaboratively train a shared global model without exposing raw local data. Blockchain technology is implemented to decentralize and replace the traditional centralized server architecture. It provides secure, transparent, and tamper-proof aggregation of model updates through a Proof-of-Stake (PoS) consensus mechanism that also incentivizes participation. Moreover, they include Maximum Mean Discrepancy (MMD) to measure the distance between each client's data distribution and the testing client, along with the L2 distance between the average global model and each local model. How much each local model should contribute during global aggregation is based on these measurements, and it improves domain adaptation and generalization on both current and new power line data. A self-incentivizing mechanism that considers both data size and model performance is also introduced through blockchain rewards to encourage sustained participation across multiple training rounds. Experimental results show that the proposed method performs better than

standard FL, FTL, centralized, and local training approaches in terms of accuracy, F1-score, and operational risk, particularly when working with newly introduced power line data and domain shifts. It enhances privacy and security; decentralized blockchain integration reduces data leakage and single-point-of-failure risks, and the reward mechanism encourages ongoing client engagement, improving model strength. However, it increases computational complexity compared to standard FL approaches, and the model remains resource-intensive, requiring significant computational resources to maintain high prediction accuracy.

Wan et al. (2023) [112] propose a ring-based decentralized federated transfer learning (RDFTL) approach for intelligent fault diagnosis in rotating machinery. This approach found key challenges in current federated transfer learning (FTL) applications, such as high communication overhead, idle waiting times between source and target clients during training, and negative transfer caused by model aggregation. The authors proposed a novel framework harnessing the bandwidth-optimal Ring-All Reduce algorithm to significantly reduce communication costs while avoiding the linear growth of overhead with the number of clients. An asynchronous domain adaptation strategy is introduced to improve training efficiency and to eliminate idle waiting by decoupling local training at source clients from feature extraction at the target client. To reduce negative transfer, a multi-perspective distribution discrepancy aggregation (MPDDA) strategy is proposed, which evaluates the diagnostic performance of local models on the target domain using three metrics: statistical distance, domain adversarial loss, and stability, which are then combined to compute the optimal aggregation weights. The result shows that the proposed method achieves rapid training speeds and produces a high-performance cross-domain fault diagnosis model while preserving data privacy. However, the method may face challenges scaling effectively in highly heterogeneous environments, and it requires additional testing in a wider variety of industrial scenarios.

This literature moves beyond standard FTL applications to address the unique security and robustness of FTL, which reveals a dynamic interplay between offensive vulnerability discovery and defensive architectural design. Zhao et al. [110] expose a critical vulnerability and demonstrate that the very process of personalization in FTL can be exploited for label inference attacks, highlighting a new attack surface. In response to the need for more resilient systems, Wang et al. [111] and Wan et al. [112] both propose FTL frameworks designed to be more secure and efficient. Wang et al. integrate blockchain as a decentralized and secure replacement for the central server by using it for tamper-proof aggregation and to incentivize participation, while Wan et al. use a highly efficient ring-based topology to eliminate central bottlenecks. These two defensive frameworks are competing architectural choices for achieving decentralized FTL, where one leverages the trust model of blockchain, while the other uses a communication-theoretic approach for efficiency. Collectively, these papers show a maturing FTL security landscape that is moving from server-centric privacy to holistic, decentralized systems that prioritize integrity and robustness, even as new, more subtle attacks are being uncovered.

6.3.3 Domain-Specific Applications of FTL in Energy and Smart Infrastructure

Wang et al. (2025) [113] propose a novel framework, Domain-Adaptive Clustered Federated Transfer Learning (DCFTL), designed to tackle key issues in Electric Vehicle (EV) charging demand prediction, including personalization, transferability, privacy preservation, and non-IID data distribution. Although all clients are initially treated equally, the authors use Maximum Mean Discrepancy (MMD) as a statistical measure to group charging stations into clusters based on similarity in their data distributions. The main forecasting model is a multi-head attention-based transformer, which is selected for its strength in capturing complex spatio-temporal dependencies in charging behavior. To solve the domain shift adaptation, the approach applies transfer learning techniques by aligning data representations from different domains within a common latent space. Maximum mean discrepancy (MMD) is used to minimize the distribution gap

between domains, facilitating smoother adaptation. A key contribution is a weighted aggregation mechanism for updating the global model; this mechanism assigns weights based on both the MMD to the cluster center and the L2 distance from the global model, thereby reducing the negative impact of dissimilar local data. To protect data, a multi-stage differential privacy method is implemented by injecting Laplace noise at the clustering, federated learning, and transfer learning stages. The key strengths of this work are its high predictive accuracy and strong transferability, especially when dealing with new or unseen data. It also reduced operational risk in comparison to traditional federated learning, centralized training, and other methods for handling non-IID data. However, relying on a geographical matrix as a proxy for traffic conditions may not fully capture real-world dynamics.

Campos et al. (2023) [114] propose a novel Federated Transfer Learning (FTL) framework designed to improve energy efficiency in smart buildings by creating accurate energy consumption prediction models while addressing critical issues of data privacy, scarcity, and system scalability. The authors identify the problem that generating such models is often infeasible due to many buildings lacking data-gathering IoT equipment, and the data that does exist raises privacy concerns, while applying traditional Federated Learning (FL) to a large number of buildings creates a debilitating communication bottleneck. To solve this, the proposed method first groups buildings into clusters based on their characteristics. From each cluster, a single representative building is chosen to act as a client in an FL environment. These few representatives collaboratively train personalized models using the Fed+ aggregation function to handle data heterogeneity. Subsequently, the trained model from each representative is transferred to the remaining buildings in its respective cluster, which then use a small fraction of their own data to fine-tune it via Transfer Learning (TL). The primary strength of this approach lies in its significant reduction of communication overhead by limiting the number of FL clients, while still allowing a large number of buildings to benefit from a collaboratively trained, privacy-preserving model. After the evaluation, it shows that the performance of TL depends on the number of samples available for each building, and when there is data scarcity, the results may vary. Also, some clusters may not benefit as much as seen in the evaluation, where not all clusters improved equally. Future work should focus on deploying and validating the solution in a real-world energy management platform.

The application of FTL to energy and smart infrastructure highlights the critical role of clustering as a foundational step for managing large-scale, heterogeneous systems. Wang et al.'s DCFTL [113] is a highly integrated framework for EV charging prediction that combines dynamic clustering (using MMD), a multi-head attention model for forecasting, and a weighted aggregation mechanism to mitigate domain shift. In contrast, the framework by Campos et al. [114] is a scalability-oriented solution designed to overcome communication bottlenecks and data scarcity in smart buildings. This leads to two distinct interpretations of FTL in practice. Wang et al. use FTL as an integrated mechanism for domain adaptation. Conversely, Mármol Campos et al. propose a pragmatic, two-stage FL+TL pipeline that uses FL among a few representative clients to solve the communication problem, and then uses TL to disseminate that knowledge to a wider population, solving the data scarcity problem. This divergence demonstrates FTL's flexibility, showing it can be implemented either as a deep, high-performance model for complex data or as a scalable, hybrid architecture for massive, resource-constrained systems.

6.4 Personalized Federated Learning (PFL)

6.4.1 Personalization Strategies and Model Adaptation in Federated Learning

Jin et al. (2023) [115] introduce pFedSD, a personalized federated learning framework that counters personalized knowledge forgetting caused by re-initializing local models from the global model in heterogeneous edge settings. pFedSD applies client-side self-knowledge distillation: each client preserves its

previous round's personalized model as a static teacher and trains the current local model (student) with a composite loss that combines task cross-entropy and a Kullback–Leibler (KL) distillation term. The KL term encourages the student's predictions to align with the teacher's, thereby preserving historical, client-specific knowledge. Empirically, pFedSD improves accuracy and fairness on non-IID benchmarks (Fashion-MNIST, CIFAR-10/100) while remaining compatible with standard FL algorithms. Its main trade-offs are sensitivity to distillation hyperparameters (distillation coefficient, temperature) and a modest per-round computational overhead to recompute teacher predictions—costs the authors argue are offset by faster convergence. Overall, pFedSD offers a practical, easily integrated solution for balancing personalization and generalization in heterogeneous edge intelligence, at the cost of distillation hyperparameter tuning and modest per-round computation for teacher predictions.

Chen et al. (2023) [116] propose pFedGate, a novel framework for efficient personalized federated learning (PFL) that addresses the dual challenges of data heterogeneity and resource constraints across clients. The core problem identified is that most PFL methods, while handling data heterogeneity, are often bottlenecked by the lowest-resource clients and incur significant overhead, limiting their practicality. To solve this, the authors introduce a structured block sparsity approach for each client. Specifically, each client is equipped with a private, lightweight, and trainable gating layer that takes a batch of local data as input and predicts continuous, block-wise gated weights. These weights are used to adjust the global model through block-wise multiplication, creating a lightweight local model that fits the client's data and device limits. This detailed, batch-level adaptation allows each client to operate its full model capacity and improve both computation and communication efficiency on account of sparsity. The strengths of this paper lie in its demonstrated ability to concurrently achieve superior accuracy guarantees for generalization, convergence, and complexity. This paper also has limitations, such as the potential for local model updates to become dense over many batches and the need to improve strength in scenarios with maximum data heterogeneity communication.

Yi et al. (2024) [117] introduce pFedKT, a personalized federated learning framework designed to mitigate performance degradation from non-IID data distributions. The method employs a dual knowledge transfer strategy: (1) a lightweight local hypernetwork generates new private models by distilling historical local knowledge, ensuring personalization, and (2) a contrastive learning objective aligns the new private model (anchor) with the global model (positive) while distancing it from its outdated predecessor (negative). This dual mechanism allows the model to simultaneously retain client-specific knowledge via the hypernetwork and absorb generalized global knowledge via contrastive alignment. While pFedKT demonstrates superior performance on non-IID benchmarks, its efficacy is highly sensitive to hyperparameter tuning, including hypernetwork architecture and the weights for its contrastive triplet loss. Furthermore, it introduces non-trivial computational overhead and, by design, may offer limited advantage in IID settings.

Wu et al. (2024) [118] introduce pFedSV, a personalized federated learning framework designed to optimize collaboration on heterogeneous (non-IID) data. Rejecting flawed model-similarity metrics used by methods like FedFomo, pFedSV operationalizes domain relevance by modeling client interactions as a coalition game. The mechanism employs the Shapley Value (SV) to quantify the precise marginal contribution of each client to a peer's personalized model performance, evaluated locally within dynamically formed coalitions. These SV scores are then repurposed as personalized aggregation weights, enabling a robust multiwise collaboration paradigm. This game-theoretic approach is shown to yield superior accuracy on non-IID benchmarks (e.g., MNIST, CIFAR-10) over baselines like FedAvg. However, this design necessitates a critical trade-off: local SV evaluation requires clients to download multiple models, significantly increasing communication overhead and creating privacy risks. The proposed mitigations, such as shared

feature extractors or adding differential privacy noise, introduce their own compromises in computation or performance.

The core challenge of PFL is to figure out how to effectively balance shared knowledge from the global model with the unique characteristics of each client's local data and move beyond simple fine-tuning to explore a diverse range of strategies. The location of personalization is key to categorizing these strategies: it can happen during the local training phase, be built into the model architecture, or occur within the aggregation mechanism. pFedSD [115] and pFedKT [117] focus on local training. pFedSD uses self-knowledge distillation to prevent "knowledge forgetting" [115], while pFedKT uses a more advanced dual-transfer mechanism with hypernetworks and contrastive learning to explicitly balance local and global knowledge [117]. In contrast, pFedGate [116] achieves personalization architecturally by learning a lightweight, sparse gating layer to adapt a shared global model with an approach highly optimized for resource-constrained devices. pFedSV [118] redefines personalization as a function of the aggregation process itself by using game-theoretic Shapley Values to create a unique set of aggregation weights for each client, ensuring they learn most from their most relevant peers The trend is a move away from simple fine-tuning towards more structured and theoretically grounded personalization, with a clear trade-off between approaches that modify the local training objective (distillation-based) and those that modify the model architecture or the aggregation process itself (gating and game theory-based).

6.4.2 Fairness, Community Structure, and Communication Efficiency in Personalized FL

Sabah et al. (2024) [119] propose FairDPFL-SCS, which is a novel framework for Fair Dynamic Personalized Federated Learning (DPFL) that addresses critical challenges in federated learning (FL), including data heterogeneity (non-IID data), fairness across clients, and inefficient client selection. The authors tackle the non-IID data challenge with the FairDPFL-SCS framework by employing a dual strategy of intelligent client selection and dynamic local model adaptation. It strategically selects a subset of clients for each training round using a fairness score that balances data diversity (via entropy-based metrics) and participation frequency (via hit-rate calculations). Each client adapts its training process based on how well it's doing, using smart learning rates and even guessed labels (pseudo-labels) to improve performance. This helps clients with little labeled data still train useful personalized models, and their updates contribute more fairly to the global model. The framework further incorporates a fine-tuning phase post-aggregation to refine model fairness and accuracy. Evaluated on datasets like MNIST, FashionMNIST, SVHN, and healthcare-related datasets (e.g., BraTS), FairDPFL-SCS outperforms baselines such as FedAvg, Ditto, and q-Fair Federated Learning (q-FFL) [120] in accuracy and fairness metrics, achieving up to 98.23% maximum accuracy with 500 clients. Computational efficiency is improved through selective client participation: training time per round drops from 130.50 s (FedAvg) to 41.25 s, though a fine-tuning step adds 12.27 s. minimum client accuracy as the network scales to 500 clients, indicating challenges in maintaining uniform performance. Furthermore, while robust against weak adversarial attacks, the model's performance degrades against stronger perturbations, and the framework could be enhanced by incorporating more advanced privacy-preserving mechanisms, as noted in their future work.

Zhao et al. (2024) [121] propose Community-Aware Personalized Federated Learning (CA-PFL) to address concept drift in social manufacturing defect detection. Assuming an honest-but-curious server, the framework models clients as nodes in a graph-structured federation social network, with edge weights defined by deep shared layer similarity (SimCKA). This graph enables the core mechanism: community detection algorithms (e.g., Louvain) cluster clients with homogeneous feature distributions, thereby mitigating concept drift. Aggregation is then personalized, as shared layers are pooled only within communities,

while clients retain personalized layers. A federation community contrastive loss (FedCCL) further accelerates convergence by maximizing intra-community model alignment. Experiments on nine datasets (e.g., NEU-CLS, TDD) demonstrate superior accuracy and faster convergence over standard PFL baselines in concept drift scenarios. The primary trade-off is the significant server-side computational burden required for continuous graph updates and similarity calculations.

Wang et al. (2024) [122] proposed a personalized federated learning framework to address communication overhead, privacy risks, and data heterogeneity by introducing a three-module client architecture (private, shared, fusion models). Only the gradients of the small shared model are uploaded, and compressed sensing is used for both compression and encryption, utilizing a chaotic encrypted cyclic measurement matrix, which functions as a lightweight key for privacy-preserving. The sparsity-adaptive iterative hard thresholding (SAIHT) algorithm further improves gradient reconstruction efficiency. Experiments on MNIST and CIFAR-10 under non-IID scenarios show nearly 15-fold reduction in communication costs, model accuracy competitive or superior to vanilla FedAvg and KD-based FL, and that accuracy drops by at most 2.57% even at high compression ratios. Visualization and label reconstruction experiments demonstrate strong resistance to gradient-based inference attacks, though the framework does not address malicious clients or poisoning attacks, focusing mainly on honest-client and inference-threat models.

These papers extend the PFL paradigm to address higher-level systemic goals to show the growing maturity of PFL, shifting research from the core question of how to personalize to the critical operational challenges of making PFL systems fair, structurally aware, and efficient. FairDPFL-SCS [119] explicitly targets fairness, using an intelligent client selection strategy that balances data diversity and participation frequency to ensure that no client is left behind. CA-PFL [121] discovers and leverages the natural community structure among clients. Instead of a single global model, CA-PFL trains a shared model for each client community, representing a middle ground between a fully global model and fully personalized models. While both approaches use clustering, FairDPFL-SCS uses it for fair client selection for a single global model, whereas CA-PFL uses it to create multiple, personalized global models. Wang et al. [122] focus on efficiency and privacy, using a novel architecture and compressed sensing to simultaneously reduce communication overhead and encrypt updates by addressing practical resource constraints. Unlike previous sections where papers often proposed competing solutions to the same problem, these contributions are largely orthogonal and complementary. A future, state-of-the-art PFL system might need to be fair (like FairDPFL-SCS), understand its internal community structure (like CA-PFL), and be communication-efficient (like the work by Wang et al. [122]).

6.4.3 Domain-Specific and Cross-Modal Applications of PFL

Xiong et al. (2024) [123] introduce a new personalized federated learning framework called Multi-task Clustering Personalized Federated Learning (MCFL). The main goal is to solve the problem of data heterogeneity in federated systems, especially for predicting carbon emissions, where data is spread out and varies significantly between sources (non-IID). MCFL works by first grouping clients with similar data (based on their model parameters) into clusters, and then creating a specific model for each group. To help these groups share useful information, the framework uses multi-task learning: the lower layers of each group's model, called "expert layers," are shared and combined with those from other groups. This allows each client to learn from a wider range of data, not just its own group. The process repeats in cycles of local training, clustering, sharing expert layers, and combining them in a personalized way. Experiments using a public dataset of monthly carbon dioxide emissions show that MCFL learns faster and predicts more accurately than standard methods like FedAvg and traditional clustering-based federated learning, as measured by Mean Absolute Error (MAE), RMSE, and Mean Absolute Percentage Error (MAPE). Overall, MCFL effectively

handles non-IID data and works well with different data distributions. However, the paper also notes that more research is needed to improve scalability and to manage clusters dynamically as data changes.

Zhou et al. (2024) [124] propose a Personalized Federated Learning with Model-Contrastive Learning (PFL-MCL) framework addressing heterogeneous, multi-modal data in human-centric Metaverse applications under an honest-but-curious threat model. The design integrates a two-stage iterative clustering algorithm for multi-center global aggregation, enabling adaptive client grouping based on dynamic local model weight changes to produce multiple personalized global models, thus improving upon traditional FedAvg inefficiencies with non-IID data. Locally, the hierarchical neural network comprises a personalized module employing hierarchical shift-window attention and a novel bridge attention mechanism for computationally efficient cross-modal fusion of high-dimensional heterogeneous inputs. The federated module incorporates Model-Contrastive Learning (MCL) with an embedding layer to accelerate convergence by minimizing divergence between local and global models while maximizing separation from previous local models, reducing communication overhead. Experiments on VSM and APR datasets demonstrate enhanced accuracy, faster convergence, and distinct clustering over state-of-the-art baselines. However, centralized coordination in multi-center aggregation introduces potential scalability and latency trade-offs in widely distributed networks. This evidences a balance between personalization benefits and system efficiency constraints in Metaverse-scale PFL deployment.

Huang et al. (2024) [125] propose a new method, personalized federated transfer learning (PFTL) framework, to predict the cycle-life of lithium-ion batteries for multiple heterogeneous clients while ensuring data privacy. The main problem tackled in this work is the difficulty of applying a deep learning-based prognostic approach in real-world industrial settings. where data is distributed across multiple institutions with significant differences in data distributions and strict privacy requirements. The authors deal with the problem of client drift and limited labeled data by first pretraining a global prognostic model on a publicly available dataset at a central server, then distributing this model to local clients for fine-tuning on their private data and finally employing a hybrid personalization strategy that combines dynamic weighted model aggregation and domain adversarial training to train models for each client. The predictive accuracy of the framework is driven by a deep learning model that features a hybrid attention mechanism, designed to capture both long-range and short-range patterns in battery data by running multi-head self-attention and multi-scale attention modules in parallel. By employing a hybrid attention mechanism that processes long- and short-term data dependencies in parallel, the proposed model achieves high predictive accuracy. This is validated by extensive experiments on four separate battery datasets, which show the method's predictions are more accurate and personalized than those from baseline centralized or standard FedAvg methods, while successfully preserving data privacy. However, the framework relies on a publicly available dataset for pretraining, which might not be accessible in many practical or safety-critical applications.

Yu et al. (2024) [126] propose a new method for Personalized Federated Continual Learning (PFCL) by implementing a multi-granularity prompt mechanism. It tackles the challenges of knowledge sharing, personalization, and Spatial-Temporal Catastrophic Forgetting (STCF) in federated learning environments. The authors observe that current methods do not effectively utilize multi-granularity knowledge representations, which are important for robust knowledge fusion and effective personalization across heterogeneous clients and evolving tasks. To solve this, the authors introduce a multi-granularity knowledge space, inspired by human cognitive processes, which decomposes knowledge into two levels. The first is a coarse-grained global prompt, learned via a shared, frozen Vision Transformer (ViT), to capture and aggregate invariant, general knowledge. The second is a fine-grained local prompt, which is class-wise and built upon the global prompt to learn client-specific details, personalizing the model, and combating temporal forgetting caused by new tasks. This method employs a selective prompt fusion mechanism on the server, aggregating only

the coarse-grained global prompts to create a generalized model, which is then distributed back to clients. The main strengths of this work include its innovative application of multi-spatial and robust handling of catastrophic forgetting. Additionally, by transmitting the small coarse-grained prompts, the method reduces communication costs and enhances privacy. However, it has some limitations remaining in its computational overhead, dependence on a pre-trained ViT backbone, focus mainly on vision tasks, and the need for a proxy dataset for server-side distillation.

Xiong et al. (MCFL) [123] and Zhou et al. (PFL-MCL) [124] use clustering as a foundational personalization strategy. However, they apply it differently. MCFL uses clustering to form groups and then shares knowledge between them via multi-task learning, while PFL-MCL creates a personalized global model for each cluster and enhances training with model-contrastive learning. Huang et al. [125] take a different approach by combining PFL with transfer learning and using domain adversarial training for personalization. It highlights how PFL can be used with other learning strategies for enhanced performance. The prompt-based framework by Yu et al. [126] confronts the temporal dynamics of Personalized Federated Continual Learning. Together, these works illustrate the advanced stage of maturation for PFL, where its techniques (clustering, contrastive learning, transfer learning) are now being customized and combined to tackle the nuanced challenges of specific, high-impact application domains, including cross-modal data and continual learning settings.

6.4.4 Mechanisms and Metrics for Quantifying & Mitigating Privacy Leakage in Personalized FL

Quantifying privacy leakage in Personalized FL requires measuring multiple complementary metrics that capture gradient reconstruction fidelity and data recovery feasibility. Three primary metrics dominate the literature: Peak Signal-to-Noise Ratio (PSNR), measuring pixel-level reconstruction quality where higher values (indicating the reconstruction is closer to the original) indicate more successful attacks (PSNR > 20 dB indicates recognizable recovery) [127,128]; Structural Similarity Index (SSIM), capturing perceptual similarity between original and reconstructed data on a 0–1 scale [127]; and Learned Perceptual Image Patch Similarity (LPIPS), measuring human-aligned perceptual distance where lower values indicate better visual recovery [128]. Membership inference attacks provide a second metric family: label reconstruction accuracy quantifies the adversary's ability to infer which data samples participated in training [94,129], while loss convergence profiling identifies clients whose models diverge from global trajectories, revealing personalization-induced leakage patterns. Practitioners must measure these metrics under realistic conditions: batch size $B \geq 8$. (reconstruction attacks fail spectacularly at B = 1 but degrade gracefully as B increases), trained rather than untrained models, and multi-round aggregation reflecting production FL where stateless single-round evaluation dramatically overestimates attack success [94].

For mitigation, the reviewed frameworks demonstrate two divergent philosophical approaches. Structural defenses re-engineer model components to break the gradient-data connection: Privacy-Encoded Federated Learning (PEFL) decomposes weight matrices into column/row representations with nonlinear transformation, achieving 20 dB PSNR reduction (rendering reconstructed images unrecognizable) while simultaneously improving model accuracy by 1.2% and reducing computational cost 2.7× [127]. In contrast, adaptive privacy mechanisms allocate protection dynamically based on measured leakage risk per round: Adp-PPFL framework quantifies risk via gradient variance and Fisher information, then allocates differential privacy budgets adaptively, adding aggressive noise (PSNR 7.44 vs. 66.56 unprotected) in high-risk early rounds where models are most vulnerable, deliberately sacrificing early accuracy for strong privacy while ultimately achieving 93% convergence on MNIST, outperforming fixed-noise approaches at 92% [95]. A third mechanism family combines cosine similarity weighting: the Pseudo-Gradient Defense weights aggregation of current and previous gradients to obscure update patterns, achieving PSNR 11.58 (MNIST)

with minimal accuracy impact [129]. For personalized FL specifically, the privacy leakage amplification from divergent client models creates two distinct attack surfaces requiring dual protection: (1) protecting shared global components via standard gradient defense mechanisms like PEFL or Adp-PPFL applied to aggregated updates, and (2) protecting personalized local parameters that uniquely characterize each client's data distribution through client-specific adaptive noise budgets proportional to local model divergence from global baseline [95]. Practitioners deploying personalized FL should employ a layered approach: first, establish baseline leakage measurement under realistic batch sizes and multi-round settings [94]; second, apply structural defenses (PEFL-style decomposition) to shared model components for maximum efficiency and utility preservation; third, overlay adaptive privacy allocation (Adp-PPFL framework) on client-specific updates, tuning privacy budget allocation to each client's personalization degree measured via parameter divergence; and finally, implement periodic privacy audits by attempting gradient reconstruction on held-out test batches to validate empirical leakage matches theoretical privacy bounds [95,127].

7 Literature Reviews on Synchronization Strategies: Synchronous and Asynchronous FL

7.1 Synchronous Federated Learning

7.1.1 Straggler Mitigation and Synchronization in Federated Learning

Lang et al. (2024) [130] propose Straggler-Aware Layer-wise Federated Learning (SALF), a new approach to address the challenge of heterogeneous edge devices causing excessive latency in synchronous federated learning (FL), known as stragglers, which often fail to complete local model updates within strict deadlines. Unlike conventional methods that ignore incomplete updates or switch to asynchronous protocols, both of which can degrade model performance under tight latency constraints, SALF takes advantage of the layer-wise nature of neural network backpropagation, letting slower clients (stragglers) submit partial gradients for the layers they finish computing before the deadline. It then updates each layer of the global model independently with potentially different sets of user contributions. The authors propose a theoretical analysis showing that SALF achieves the same long-term convergence rate as standard federated learning without timing constraints. On MNIST with Multilayer Perceptron (MLP) and CNN, SALF maintains comparable accuracy to vanilla FL when up to 90% of users are stragglers (test accuracy 0.81 for MLP, 0.90 for CNN), compared to drop-stragglers' severe degradation (0.49 and 0.28, respectively), and on MNIST with CNN models achieves only 5% accuracy loss vs. 90% for drop-stragglers at tightest latency constraints. Convergence slows modestly for deeper networks due to increased variance, and the uniform backpropagation depth assumption may not reflect real heterogeneous edge devices, though SALF maintains simplicity and can combine with other FL optimizations.

Liu et al. (2024) [131] propose a novel buffer-aided synchronous federated learning (FL) scheme designed to operate over wireless networks. The central problem addressed is the "straggler effect," where the overall speed of synchronous FL is limited by the slowest participating wireless device due to heterogeneous computing and communication capabilities. The authors propose a solution: a buffer at each wireless device to temporarily store collected data. It allows the amount of data used for local training to be adaptively adjusted based on the device's specific resources. The approach ensures that all devices complete training and upload their updates together, which in turn accelerates model aggregation. The proposed method is formalized as a delay-aware online stochastic optimization problem that aims to minimize the long-term average "staleness" of the training data. After all, a metric was introduced to represent data freshness. Moreover, it incorporates the Entropic Value-at-Risk (EVaR) to manage the risk of discarding useful data, which ensures the stability of the data queues at each device. The scheme formulates a delay-aware Lyapunov optimization problem minimizing long-term average staleness while bounding queue stability, with distributed resource allocation for power, bandwidth, and training data size across devices. Hardware-in-the-loop simulations on Received

Signal Strength Indicator (RSSI) (5×10^3 samples) and Bit Error Rate (BER) (5×10^5 samples) datasets show buffer-aided synchronous FL achieves convergence with fewer rounds and lower staleness than no-buffer baselines; when 8 devices are used, training time decreases substantially, and the proposed scheme becomes faster than asynchronous FL when the number of devices exceeds 6. Accuracy improves with more devices and smaller epoch counts; the approach handles heterogeneous channel conditions and dynamic stragglers effectively but assumes sufficiently large buffers and relies on FIFO discarding, which may not be optimal for all edge scenarios.

Rajesh et al. (2025) [132] propose a detailed comparison between synchronous and asynchronous federated learning (FL) approaches, which evaluate the trade-offs between model accuracy and convergence speed. The problem it addresses is the lack of clear, quantitative guidance for professionals on when to choose one training mode over the other, since both have inherent strengths and weaknesses. Synchronous FL can be delayed by slow clients (the "straggler effect"), while asynchronous FL can lead to "stale updates" that may reduce model accuracy. The authors tackle this problem using well-structured experimental setups in a simulated environment with 100 clients. To consider model complexity, they use the MNIST dataset with four different neural network designs. They systematically vary key parameters, such as data distribution (IID vs. non-IID), batch size, the number of clients, and learning rate, to measure their effects on accuracy and convergence epochs for both FL and asynchronous FL (AFL). The results consistently show that while AFL converges faster (e.g., reducing convergence epochs by an average of 15.92% in the default setup), synchronous FL achieves higher accuracy (improving it by an average of 6.68%). The paper offers a thorough quantitative analysis that highlights key differences between AFL and FL, for example, AFL becomes faster as the number of clients increases, while FL maintains stronger accuracy in non-IID data settings. However, the study is based only on one relatively simple dataset (MNIST) and a standard aggregation algorithm (FedAvg), so the results may not generalize to more complex data or advanced FL frameworks.

The reviewed literature explores the straggler issue from two angles: proposing novel mitigation strategies and quantifying the fundamental trade-offs that motivate them. The papers by Lang et al. [130] and Liu et al. [131] offer two distinct and complementary philosophies for managing stragglers. SALF proposes a server-side adaptation strategy with modified aggregation logic to accept partial, layer-wise model updates. In contrast, the buffer-aided scheme by Liu et al. proposes a client-side adaptation that dynamically adjusts each client's training workload to ensure all participants can finish a complete update simultaneously. Lang et al. adapts the nature of the update, while Liu et al. adapt the volume of work. These two straggler mitigation techniques are complementary: a system could use buffer-aided workload adjustment to reduce the number of stragglers, and then use SALF's layer-wise aggregation to salvage useful information from any clients that still fail to meet the deadline. The importance of these mitigation strategies is underscored by the analysis from Rajesh et al. [132], which quantifies the trade-off and accredits that synchronous FL generally yields higher accuracy while asynchronous FL converges in fewer rounds.

Across all three studies, the results reveal distinct operating regimes: SALF sustains near-FedAvg accuracy even under extreme straggler pressure, retaining 0.81 with MLP and 0.90 with CNN on MNIST at 90% stragglers vs. drop-stragglers collapsing to 0.49 and 0.28, and under the tightest latency it loses only 5% accuracy compared to losses of 15% for AsyncFL, 70% for HeteroFL, and 90% for drop-stragglers [130]. In contrast, the buffer-aided synchronous scheme cuts total computation time relative to synchronous without buffers for all device counts and overtakes asynchronous FL only when the number of devices exceeds about 6, while its Lyapunov–EVaR design reduces long-term average staleness g[a(n)] as device count grows in limite-data settings, indicating fresher updates in practice [131]. Complementing these results, the cross-mode study quantifies that asynchronous FL converges in fewer epochs by an average of 15.92% in the default setup, with the speedup rising from 10.5% to 15.92% as clients increase from 25 to 100, whereas synchronous

FL achieves a higher final accuracy by 6.68% on average overall, with margins of 4.49%, 4.54%, and 6.68% at 25, 50, and 100 clients, and a similar 6.72% advantage under non-IID data; these patterns persist across learning rates where AFL's epoch reduction ranges from 13.92% at low LR to 15.92% at high LR while FL's accuracy edge remains about 6.6%–6.7% [132]. In practice, this suggests using buffer-aided workload control to shrink per-round times and surpass asynchronous only beyond roughly six devices [131], applying SALF to recover about 0.32–0.62 absolute accuracy over drop-straggler baselines at very high straggler rates [130], and relying on asynchronous updates when a roughly 14%–16% reduction in convergence epochs outweighs a consistent 6%–7% accuracy gap relative to synchronous aggregation [132].

7.1.2 Semi-Synchronous and Adaptive Scheduling Frameworks

Jmal et al. (2025) [133] introduces TUNE-FL, an adaptive semi-synchronous, semi-decentralized Federated Learning (FL) framework addressing client heterogeneity—differences in computational resources and data distributions. The framework solves how to use FL efficiently across heterogeneous clients and multiple edge servers while avoiding single-point failures, communication slowdowns, and delays from varying computational speeds. TUNE-FL first ensures consensus among edge servers regardless of network topology by determining the minimum message-passing rounds required for information sharing. Its central mechanism is an adaptive semi-synchronous training process where clients stop when they finish training or when time runs out, each estimating convergence time for the next round. These estimates are aggregated using the interquartile mean (IQM) to calculate the next synchronization deadline, filtering outliers from very slow or unusually fast devices. The framework proportionally down-weights clients that miss the round deadline and still incorporates their partial updates which ensures that their progress contributes without degrading overall training dynamics. On UNSW-NB15 and CIC-IDS2017, TUNE-FL attains accuracy of 0.862 and 0.979 with F1-scores of 0.948 and 0.937, respectively, surpassing FedAvg-SDFL, SemiSync-SDFL, and SD-FEEL while cutting total training time by roughly 92–97× through interquartile-mean–driven synchronization deadlines that curb the impact of stragglers. Staleness tolerance remains strong under non-IID data and dynamically bounded delays, yielding fast, stable convergence across heterogeneous devices and topologies, although the current evaluation is limited to intrusion detection workloads and does not explicitly model communication latency on lossy channels.

Yu et al. (2024) [134] introduce DecantFed, a new federated learning (FL) approach that sits between traditional synchronous and asynchronous methods, making it better suited for real-world IoT systems where devices have different speeds and network conditions. The authors tackle challenges such as the straggler problem, high communication costs, and model staleness by introducing a dynamic client clustering mechanism, grouping clients into tiers based on their computing and communication latencies so that clients in different tiers upload their local models at distinct frequencies. In this semi-synchronous FL framework, the FL server waits for model uploads from clients in each tier according to their tier-specific deadlines, which helps balance client participation and communication cost. DecantFed jointly optimizes client clustering, bandwidth allocation, and local training workloads to maximize the number of training data samples processed per unit time, using the heuristic joint client clustering and bandwidth allocation algorithm, for efficient tier assignment and bandwidth distribution. Bandwidth is allocated to each tier using Frequency Division Multiple Access (FDMA), and within each tier, clients share bandwidth via Time Division Multiple Access (TDMA). Dynamic workload optimization is performed using linear programming, allowing clients with higher computational capacity to process more data samples, and the global model aggregates all participating clients' updates with larger influence for those processing more data. Higher-tier clients use larger learning rates to mitigate staleness, and loss clipping prevents divergence under non-IID data. In simulations on MNIST and CIFAR-10 under IID and non-IID settings, DecantFed

matches or slightly exceeds FedAvg's final accuracy while converging much faster (e.g., on MNIST it reaches 90% by 130,000 s vs. FedAvg's 45%/73% at 200,000 s; on CIFAR-10 it attains 70% non-IID and 73% IID vs. FedProx's 39%/45%, and achieves 41% accuracy in 50,000 s vs. 200,000 s for FedAvg), yielding at least a 28% accuracy gain over FedProx and showing clear deadline sensitivity (e.g., non-IID accuracy 69.07% at 2.5 s, 73.48% at 10 s, 72.61% at 80 s), with dynamic learning rates and tiered deadlines providing robust staleness and bounded-delay handling but adding server-side complexity and assuming labeled data.

Roy et al. (2025) [135] introduce a new way for hospitals to work together on AI models using medical images, while keeping patient data safe, where hospitals can update their AI models whenever it suits them but also have regular times to sync up, reducing waiting times and accelerating secure sharing via chaos-based encryption using the Henon Logistic Crossed Couple Map (HLCML) and CNN-based training without sharing raw data. It keeps both medical images and model weights safe by encrypting them end to end before cloud upload, aggregating encrypted updates using identical random numbers from a central key hub, and combining DP-SGD with gradient clipping and Gaussian noise under a privacy budget that reports a noise multiplier $\epsilon = 0.25$ in experiments, alongside adaptive learning-rate scheduling to handle heterogeneous, non-IID datasets and semi-synchronous rounds to curb stragglers and balance communication and computation. Empirically, the framework reaches an average 94.3% accuracy with MobileNetV2 on non-IID medical datasets, achieves over 85% convergence within 100 communication rounds, and limits per-round computation to 0.0143 s, while HLCML PRNG and confusion–diffusion steps add 0.00089 and 0.0054 s, respectively; compared to PPFLHE, PEMFL, and DP-SGD, it improves accuracy to 87.4% vs. 83.2%–80.9%, cuts computation time to 0.0143 s vs. 0.084–0.045 s, and reduces privacy leakage to 0.5% vs. 1.7%–3.2%. Robustness under privacy attacks shows higher Signal-to-Noise Ratio (SNR) and lower RMSE than baselines across multiple cancer image sets, with FID rising due to stronger obfuscation, and convergence remains stable for modest noise ($0 < \epsilon \leq 0.25$), though there is no explicit staleness/latency ablation and bounded-delay behavior is inferred rather than stress-tested on real networks, indicating promising efficiency and privacy with pending validation in cross-hospital deployments and larger-scale, delay-aware benchmarks.

The works are based on a shared conclusion that the rigid difference between synchronous and asynchronous FL is often suboptimal. Instead, they explore the possible middle ground of semi-synchronous and adaptive frameworks. However, these papers reveal that semi-synchronous is not a single concept but a design space with diverse strategies. Jmal et al. (TUNE-FL) [133] and Yu et al. (DecantFed) [134] propose adaptive systems that use different mechanisms. TUNE-FL uses a globally adaptive deadline by dynamically calculating a synchronization deadline for each round based on client-provided convergence estimates. DecantFed groups clients into tiers based on their latency, assigning different, fixed synchronization frequencies to different client capability tiers. The framework by Roy et al. [135] highlights a different priority: co-designing a pragmatic semi-synchronous protocol with a robust security stack for a high-stakes domain. The overarching trend is a move towards highly adaptive synchronization. These advanced frameworks listen to the state of the network and adjust accordingly. Rather than forcing heterogeneous clients into a single, rigid protocol. This demonstrates that the future of efficient FL synchronization lies in dynamic, intelligent systems that can adapt their coordination strategy in real time.

Quantitative comparisons across TUNE-FL, DecantFed, and Roy et al.'s semi-synchronous framework reveal complementary design choices that optimize different performance dimensions [133–135]. TUNE-FL achieves an F1-score of 0.948 on UNSW-NB15 and 0.937 on CIC-IDS2017 while reducing training duration by approximately 92–97 times relative to fully synchronous baselines (FedAvg-SDFL and SemiSync-SDFL), translating to convergence in under 100 s for intrusion detection tasks [133]. DecantFed, tested on MNIST and CIFAR-10, demonstrates that optimized client clustering and workload assignment yield

28%–70% accuracy improvements over FedProx (e.g., 41% accuracy in 50,000 s vs. 200,000 s for FedAvg-style training on MNIST, and 70%–73% non-IID accuracy on CIFAR-10 within 130,000 s compared to FedProx's 39%–45%); critically, its deadline sensitivity shows non-IID accuracy ranges from 69.07% at 2.5-s deadlines to 73.48% at 10-s deadlines, indicating that tier-based frequency allocation fundamentally rebalances the straggler-staleness trade-off [134]. Roy et al.'s chaos-encrypted framework, operating on medical imaging with MobileNetV2, achieves 94.3% non-IID accuracy with per-round computation limited to 0.0143 s; compared to privacy baselines (PPFLHE, PEMFL, DP-SGD), it cuts computation time from 0.084–0.045 s to 0.0143 s while improving accuracy to 87.4% vs. 83.2%–80.9%, though its privacy cost remains higher (0.5% vs. 1.7%–3.2% leakage) [135]. These results collectively indicate that TUNE-FL's interquartile-mean adaptive deadlines excel at heterogeneous device synchronization through statistical outlier filtering [133], DecantFed's tiered frequencies optimize throughput per unit time by assigning high-capacity clients elevated upload rates that better utilize their compute surplus [134], and Roy et al.'s integration of differential privacy with semi-synchronous coordination sacrifices per-round efficiency to guarantee formal privacy guarantees [135], suggesting practitioners should select frameworks based on whether straggler-mitigation (TUNE-FL), throughput-maximization (DecantFed), or privacy-criticality (Roy et al.) is the primary constraint.

7.1.3 Communication and Computation-Efficient Federated Learning

Zhou et al. (2024) [136] propose a novel three-layer Federated Learning (FL) framework called Parameter Selection and Pre-synchronization Federated Learning (PSPFL). It is designed to train a model and improve accuracy in mobile edge-cloud networks. In this model, clients have limited computation and communication resources. The main problem found is the high communication cost and inefficiency in traditional FL, which requires all clients to upload all model parameters, even if some are not important. This leads to slow convergence and wastes resources. To solve this issue, the authors propose a three-layer system where clients selectively transmit only significant model parameters, the ones that help most with global convergence, to their base stations, which then perform parameter pre-synchronization and combine before forwarding results to the central server. The framework solves this issue using a Deep Q-Network with Alternating Minimization (DQNAM), which finds the most suitable count of local training and pre-synchronization rounds. It first uses the Alternating Minimization (AM) algorithm for initialization, and then applies DQNAM for dynamic optimization. By reducing training loss without adding extra time, this approach improves efficiency. Experiments with various datasets and models reveal that PSPFL decreases the total FL completion time and training loss by 20.72% to 69.25% compared to the baseline. This proves its strength in communication efficiency, scalability, and adaptability. However, the joint optimization problem is a nonlinear integer programming problem, which is Non-deterministic Polynomial-time hard (NP-hard) and requires sophisticated algorithms (like DQNAM) to solve. And it requires careful tuning to avoid negative impacts on model accuracy when reducing transmitted parameters.

Jiang et al. (2024) [137] propose FedMP, a novel federated learning (FL) framework designed to improve both computation and communication efficiency in distributed machine learning, especially when dealing with heterogeneous edge devices; the main problem addressed are the challenges of resource overhead and inefficiency in traditional FL systems, where training large models on resource-limited devices causes high computation and communication overhead and straggler bottlenecks, and FedMP solves this by having the parameter server assign customized, smaller sub-models via adaptive pruning using an online Extended Upper Confidence Bound (E-UCB) multi-armed bandit to determine per-worker pruning ratios without prior device knowledge, while a Residual Recovery Synchronous Parallel (R2SP) scheme reconstructs full model structure before aggregation to avoid accuracy loss, with extensibility to peer-to-peer settings. Experiments on 30 Jetson TX2 devices across CNN/MNIST, AlexNet/CIFAR-10, VGG-19/EMNIST, and

ResNet-50/Tiny-ImageNet show up to 4.1× end-to-end speedup over baselines, 2.2× faster than synchronous FL on AlexNet/CIFAR-10 to 80% accuracy (10,906 s vs. 24,017 s), and 2.0×–1.1× faster than FedProx and 1.8×–1.2× than FlexCom across models, with accuracy matching synchronous FL at fixed time and improved accuracy under fixed time budgets due to shorter rounds and more iterations. FedMP remains robust under non-IID data, cutting completion time by 30%/23%/16%/12% vs. Syn-FL/UP-FL/FedProx/FlexCom for VGG-19 on EMNIST at non-IID level 30, scales to 30 workers with 2.4× speedup vs. Syn-FL, and reduces per-round completion-time variance by 58.1%–73.8% via E-UCB, indicating effective straggler mitigation; algorithm overhead is negligible relative to compute/communication, though performance degrades with overly coarse pruning granularity θ and real-world staleness or bounded-delay dynamics are not explicitly stress-tested beyond synchronous rounds and Peer-to-Peer (P2P) simulations, suggesting future delay-aware benchmarks for convergence under network variability.

Comparative analysis across PSPFL and FedMP reveals divergent optimization strategies targeting different performance bottlenecks in communication-computation trade-offs [136,137]. PSPFL employs parameter selection with pre-synchronization across a three-layer edge-cloud architecture [136], achieving a combined reduction in FL completion time and training loss of 20.72%–69.25% compared to FedAvg, FedCS, FedProx, and the Alternating Minimization baseline. Specifically, on MNIST with CNN, PSPFL reduces transmitted parameters by 38.11% at three clients and sustains roughly 35%–36% as clients scale to seven, with similar trends on Fashion-MNIST and CIFAR-10, indicating selective transmission preserves scalability [136]. In contrast, FedMP targets computation heterogeneity via adaptive pruning guided by an Extended UCB bandit, achieving up to 4.1× end-to-end speedup over baselines [137]. On AlexNet/CIFAR-10, it reaches 80% accuracy in 10,906 s vs. 24,017 s for synchronous FL (2.2×) [137]. It also improves 2.0–3.6× over FedProx and 1.1–1.8× over FlexCom across models, and at non-IID level 30 cuts completion time by 30%, 23%, 16%, and 12% vs. Syn-FL, UP-FL, FedProx, and FlexCom for VGG-19/EMNIST [137]. R2SP recovers pruned parameters pre-aggregation in FedMP, whereas PSPFL's DQNAM jointly optimizes local and pre-sync rounds [136,137]. PSPFL excels under bandwidth constraints; FedMP dominates when computational heterogeneity is the primary bottleneck [136,137].

Table 5 gives an overview of recent synchronous federated learning algorithms and shows how each method handles staleness, delay, convergence behavior, datasets, client scale, and core performance results.

Table 5: Benchmark of synchronous federated learning algorithms

Algorithm /Ref.	Staleness handling	Delay (τ)	Convergence	Dataset/Task	Client scale	Key metrics (Accuracy, Rounds, Comm.)
SALF [130]	Layer-wise partial gradients	Up to 90%	Same as FedAvg $O\left(\frac{1}{T}\right)$	MNIST (MLP, CNN), CIFAR-10 (VGG)	Tens (30 sim.)	Accuracy 0.81 (MLP), 0.90 (CNN) at 90% stragglers; rounds comparable; no extra communication.
Buffer-aided Sync [131]	Adaptive workload via buffer	Bounded	Faster than no-buffer baseline	RSSI, BER; wireless sim.	6–8 devices	Reduced training time; minimized staleness; slight comm. overhead

(Continued)

Table 5 (continued)

Algorithm /Ref.	Staleness handling	Delay (τ)	Convergence	Dataset/Task	Client scale	Key metrics (Accuracy, Rounds, Comm.)
TUNE-FL [133]	IQM-based semi-sync deadline, weighted updates	Dynamic	Stable, robust to non-IID	UNSW-NB15, CIC-IDS2017	100/10 servers	F1 0.86–0.98; 92–97× faster training; dynamic comm.
DecantFed [134]	Dynamic tier clustering	Tiered	Strong empirical	MNIST, CIFAR-10 (IID/non-IID)	100 clients	90% MNIST, 70% CIFAR-10; up to 4× faster; optimized comm.
Chaos-Encrypted Semi-Sync [135]	Encrypted semi-sync; adaptive LR	Inferred	Empirical with DP	Medical images (cancer)	Few hospitals (sim)	Acc. 0.943 non-IID; 85% conv. in 100 rounds; low compute; encryption overhead
PSPFL Param. Select. [136]	Pre-sync with parameter selection	Full sync	Theoretical bounds	MNIST, Fashion MNIST, CIFAR	Small clusters (9 clients)	20%–69% faster; transmission cut 35%–38%; faster conv.
FedMP Adaptive Prune [137]	Per-client pruning	Sync	Comparable accuracy	CIFAR-10, EMNIST, Tiny-ImageNet	30 device testbed	Comparable accuracy; up to 4.1× speedup; reduced communication cost

7.2 Asynchronous Federated Learning

7.2.1 Asynchronous Aggregation and Staleness Compensation

Miao et al. (2024) [138] propose AsyFL, an asynchronous federated learning scheme that uses time-weighted and stale model aggregation to address poor model performance caused by device heterogeneity and frequent client dropouts. To further protect client privacy with minimal overhead, the authors introduce Asy-PPFL, which integrates symmetric homomorphic encryption (SHE) into AsyFL. Traditional synchronous FL methods assume all clients are always available and have IID data, which leads to low accuracy and increased communication time when clients drop out, while existing asynchronous or privacy-preserving approaches often either ignore privacy risks or introduce significant computation and communication overheads. To address these challenges, AsyFL uses a time-weighted stale model aggregation (SMA) method that prioritizes recent client updates, gives more weight to it and less to older updates. By doing this, even updates from delayed or offline clients can be included, helping the global model stay

accurate and robust, even when up to 80% of clients are stragglers. Building on AsyFL, Asy-PPFL incorporates symmetric homomorphic encryption (SHE) to achieve lightweight privacy preservation by allowing clients to quantize and encrypt their local model updates before uploading, thus enabling efficient and secure aggregation without the high computation and communication overhead of asymmetric schemes like Paillier or CKKS. The paper provides a theoretical analysis showing that Asy-PPFL is indistinguishable under known plaintext attacks and proves the convergence of both AsyFL and Asy-PPFL. Experiments on MNIST, Fashion-MNIST, and CIFAR-10 show that AsyFL and Asy-PPFL maintain accuracy comparable to FedAvg, even achieving 58.40% and 58.26% on CIFAR-10 with 80% stragglers, while reducing communication overhead by more than 10× compared to Cheon-Kim-Kim-Song (CKKS) based privacy-preserving federated learning methods. However, the paper assumes that all clients are honest-but-curious and use the same network structure and parameters, and does not address scenarios with malicious clients.

Zhu et al. (2023) [139] propose an asynchronous federated learning framework that mitigates staleness with a lightweight compensation algorithm and speeds convergence via UCB-based client selection for wireless edge networks. The staleness compensation includes the first-order term of the gradient's Taylor expansion, approximated by an element-wise product with a tuning parameter λ to avoid costly Hessian computation. Client selection is cast as a multi-armed bandit problem using an upper-confidence-bound strategy and a fixed-deadline policy to prefer low-latency clients. Evaluated on MNIST with a standard CNN, the method yields faster convergence, higher test accuracy, and lower training latency than baselines (notably at 600 and 1000 s) while keeping computation and storage costs low. Trade-offs include sensitivity to the λ tuning and the lack of deployment results in real wireless networks, so practical challenges may remain.

Hu et al. (2023) [140] propose an asynchronous federated learning (FL) framework with periodic aggregation, featuring channel-aware and data-importance-based scheduling, to address the straggler issue and communication inefficiency in wireless networks. The key problem addressed is the degradation of convergence performance in synchronous FL caused by the straggler issue arising from heterogeneous device computing capabilities, non-independent and identically distributed (non-IID) data, and limited wireless communication resources. The authors propose a device scheduling policy for asynchronous federated learning that jointly optimizes for wireless channel quality and the statistical representativeness of local data, thereby reducing the bias and variance in aggregated model updates by selecting devices with strong link conditions and diverse data. The authors introduce an age-aware aggregation strategy, where the contribution of each device's update is weighted according to its Age of Local Update (ALU), thereby emphasizing more recent updates to reduce errors from asynchronous participation. The framework's theoretical convergence analysis explicitly accounts for practical system constraints, including model update sparsification, quantization, and intra-iteration asynchrony and demonstrates that the proposed approach achieves a reduced optimality gap under these conditions. However, optimal values for parameters such as the aggregation period and the number of scheduled devices are not analytically derived and must be empirically tuned for different system settings.

Xu et al. (2023) [141] propose FedLC, an asynchronous federated learning (FL) framework for edge computing (EC) that speeds up model training by allowing devices to share gradients directly with each other, especially when their data distributions differ. FedLC uses a demand-list to efficiently choose which devices should collaborate, and it assigns adaptive learning rates to each device to reduce the negative effects of slow devices (synchronization barriers) and outdated updates (model staleness). To address these challenges, FedLC builds on an asynchronous federated learning model by adding a local collaboration step. In this approach, each edge device not only sends its local model updates to the central server but also shares its gradient directly with a few other devices that have different data distributions. Each device aggregates gradients from both its own data and the gradients received from collaborating neighbors, improving model

generality and reducing the impact of non-IID data. The process of choosing which devices to collaborate with is managed by a server-side "demand-list," which is dynamically updated based on estimated benefits and resource constraints and turns the usual pull-based requests for gradients into push-based sharing, eliminating blocking and waiting. FedLC is specifically designed to minimize global loss while respecting the communication and computation constraints of edge devices, and it reduces communication overhead by allowing each device to collaborate with only a limited set of other devices, as determined by resource budgets and the demand-list mechanism. Additionally, to handle the problem of model staleness caused by device heterogeneity, the paper introduces SC-FedLC (Staleness-Compensated FedLC), which assigns adaptive learning rates to devices based on how often they participate in global updates. Experimental evaluations on real-world datasets such as CIFAR-10, EMNIST, and Image-100 show that FedLC outperforms baselines like FedAvg, CE-AFL, and Asynchronous Federated Optimization (AFO) in accuracy and completion time, particularly under high device heterogeneity. While the mechanism reduces network traffic compared to some asynchronous methods by being more efficient, it can still consume more traffic than the conventional synchronous FedAvg, as it introduces additional communication for local collaboration.

Under high straggler pressure and privacy constraints, AsyFL sustains CIFAR-10 at 58.40% accuracy with 80% stragglers while Asy-PPFL reaches 58.26% and cuts communication by over 10× vs. CKKS-based PPFL with about 3× faster encryption and roughly 2× lower total crypto time, making it quantitatively strongest when dropout and privacy coincide [138]. Zhu's first-order Taylor compensation paired with UCB client selection achieves higher test accuracy at fixed budgets of 600 and 1000 s and accumulates more rounds within the same time by biasing toward faster responders without channel state information (CSI), translating into faster time-to-accuracy under realistic wireless variability [139]. When communication must be conserved and fully asynchronous updates fluctuate, Hu's periodic aggregation yields higher test accuracy than FedAsync and FedAvg under equalized resource budgets for both i.i.d. and non-i.i.d. data, with the best setting observed at an aggregation period near $\tilde{T}$ = Tmax/4 and a measurable trade-off in the scheduled fraction R between compression loss and bias [140]. For non-i.i.d. workloads, FedLC's local collaboration reduces traffic to reach targets and speeds convergence, requiring 81.4 GB vs. 87.3 GB for CE-AFL to hit 0.3 accuracy on Image-100 and 71.7 GB with SC-FedLC, while timeout retransmission improves accuracy by about 7% at equal time, for example around 0.53 vs. 0.51 on CIFAR-10 at 1700 s, indicating superior communication-to-accuracy efficiency and staleness resilience in edge settings [141]. Collectively, the quantitative evidence favors AsyFL or Asy-PPFL for extreme straggler plus privacy regimes, Zhu's compensation for latency-aware acceleration without CSI, Hu's periodic aggregation for accuracy under tight uplink budgets, and FedLC when non-i.i.d. generality and lower bytes-to-target-accuracy dominate system objectives [138–141].

The challenge of model staleness in asynchronous federated learning (AFL) has led to two distinct paths. One does not try to eliminate stale updates. It accepts them as part of the system's nature, and adjusts accordingly: AsyFL [138] reduces their influence by scaling gradients with observed delay, while Hu et al. [140] encode this intuition through the "Age of Local Update," a metric that modulates aggregation weights based on how long a client's update has been pending. The second approach does not just dampen stale updates; it tries to fix them. Zhu et al. [139] use Taylor series to guess what those gradients would've looked like if training had kept going on the latest model; in practice, that means estimating where they're pointing now. But both strategies still wait for delays to happen before reacting. A more promising trend is emerging: proactive staleness avoidance through intelligent client selection, where systems prioritize clients likely to produce fresh, informative updates before delays accumulate [139,140]. Building on this, FedLC [141] breaks from server-centric paradigms entirely, enabling direct peer-to-peer gradient exchange among clients. This architectural innovation creates localized convergence loops that accelerate learning, reduce staleness at

its source, and are especially effective under non-IID data distributions, where slow server feedback typically deepens divergence. Collectively, these works reveal that AFL is no longer merely about handling unordered updates; it is evolving into a class of systems that anticipate, adapt, and even restructure how learning flows across the network.

7.2.2 Semi-Asynchronous and Adaptive Frameworks

Chen et al. (2025) [142] propose an adaptive semi-asynchronous federated learning (ASAFL) framework for wireless networks that dynamically controls the synchronous degree and calibrates global updates using cached historical client gradients to counter stragglers, delayed gradients, and non-IID bias while explicitly balancing latency and accuracy. The server aggregates newly received gradients with stored ones from absent clients and jointly optimizes the per-round synchronous degree and bandwidth via a Lyapunov-based online policy minimizing a drift-plus-penalty objective. Theoretical analysis shows that allocating more time to later rounds to increase the synchronous degree reduces convergence error under a fixed training horizon. On MNIST and CIFAR-10, ASAFL improves accuracy by 3%–8% and achieves 1.6×–4.3× faster time-to-target compared to SAFL, SAFL-SC, and asynchronous baselines (e.g., 88% in 9.95 s vs. 43.09 s on MNIST with $\alpha = 0.01$, and 65% in 0.97×10^4 s vs. 1.63×10^4 s on CIFAR-10 with $\alpha = 0.01$). The method intentionally trades higher latency from larger synchronous degrees for improved convergence and remains validated only on MNIST and CIFAR-10, limiting its generality.

Sha et al. (2024) [143] present EEFL, a three-tier client–edge–cloud federated learning (FL) framework operating under standard privacy-through-locality assumptions. EEFL integrates cross-layer semi-asynchronous client–edge updates with synchronous edge–cloud aggregation to reduce idle time and communication latency while mitigating Non-IID-induced staleness. The framework pairs clients to edges by computing cosine similarity over singular values obtained from the singular value decomposition (SVD) of selected local model layers, intentionally grouping dissimilar clients per edge so that each cluster better approximates the global distribution and accelerates convergence. EEFL further partitions models into shared and personalized layers and introduces a personalized model-contrastive loss that aligns shared local–global representations while stabilizing personalized heads through an exponential moving average (EMA) of historical local models, thereby mitigating drift from asynchronous updates. Experiments on MNIST, CIFAR-10, and FashionMNIST demonstrate speedups between 2.5× and 7.0× alongside higher accuracy compared to hierarchical baselines under comparable resource conditions. These improvements, however, trade additional client computation (from SVD and contrastive terms) and architectural rigidity for faster convergence and reduced communication traffic, with fairness and adversarial robustness remaining unaddressed.

Zhou et al. (2024) [144] propose an adaptive segmentation-enhanced Asynchronous Federated Learning (AS-AFL) framework to address scalability, reliability, and network heterogeneity challenges in large-scale, decentralized intelligent transportation systems (ITS) by leveraging meta-learning-based client grouping and hybrid synchronous-asynchronous model aggregation. The authors address the inefficiency and instability of federated learning (FL) in large-scale, dynamic vehicular networks, where traditional centralized FL suffers from synchronization bottlenecks, vulnerability to single points of failure, and limited adaptability to heterogeneous, network-agnostic environments by proposing a decentralized, meta-learning-based adaptive segmentation and hybrid aggregation approach. The first key component is a meta-learning-based adaptive segmentation technique, which automatically organizes client nodes such as vehicles into edge groups by analyzing and grouping them according to shared attributes like network performance, device characteristics, and the types of models they use. This meta-learning process employs a double-layer loop (inner and outer) to optimize segmentation parameters, enabling the system to quickly adapt to new tasks and changing

environments by training on a variety of scenarios, essentially allowing it to "learn how to learn." Second, the framework introduces an integrated aggregation mechanism where, within each group, nodes perform synchronous model updates with each other in a peer-to-peer manner using a gossip protocol (horizontal FL), while between groups, model updates are shared asynchronously (vertical FL). This hybrid aggregation strategy improves both the efficiency and reliability of learning in large, dynamic transportation networks. The authors conducted the experiment on an open-source EMNIST dataset with over 1000 nodes, where AS-AFL performed better than conventional FL methods such as S-FedAvg and A-FedAvg, and achieved faster convergence (542 s for 200 iterations vs. 957 s for A-FedAvg), higher accuracy (93% vs. 88%), and more than 90% reduction in Wide area network (WAN) communication overhead. However, the paper is evaluated in a simulated environment, which might limit the generalizability of the results in a real-world ITS deployment.

ASAFL demonstrates 2.9% accuracy gain on MNIST and 5.78% on CIFAR-10 over benchmark schemes such as SAFL and Asynchronous FL, achieving 2.1× and 1.9× speedup to target accuracies, respectively, while dynamically adjusting the synchronous degree per round via Lyapunov optimization; its gradient calibration mechanism incorporates historical gradients from non-participating clients, directly counteracting non-IID bias at the cost of higher per-round overhead [142]. EEFL achieves 2.5–7.0× speedup over baselines like FedAvg, MOON, and FedProx with top accuracies of 0.9446 on MNIST (30 rounds to 90% target), 0.862 on CIFAR-10 (29 rounds to 85% target), and 0.8736 on Fashion-MNIST (31 rounds), using crosslayer semi-asynchronous aggregation in a three-tier client-edge-cloud architecture paired with SVD-based cosine similarity clustering that ensures each edge covers maximally dissimilar clients [143]. AS-AFL, tested on EMNIST with over 1000 nodes, converges in 542 s to 93% accuracy for 200 iterations vs. 957 s for A-FedAvg at 88%, cutting WAN communication overhead by over 90%; its meta-learning-based adaptive segmentation into homogeneous groups enables intra-group synchronous gossip aggregation while maintaining inter-group asynchronous updates, yielding 94.03% accuracy with superior convergence smoothness [144]. Critically, ASAFL's adaptive synchronous degree control trades earlier-round speed for later-round accuracy through ascending patterns (improving by 2%–5% over fixed-degree schemes), EEFL's dissimilar-client pairing maximizes local diversity within edge clusters (outperforming HiFlash random pairing by 1.5%–5%), and AS-AFL's meta-learned grouping avoids the CSI collection overhead that HiFlash requires, making it safer and more scalable to 1000+ nodes [142–144]. These results collectively suggest selecting ASAFL when Lyapunov-optimized synchronous degree allocation is feasible and non-IID data dominance is critical, EEFL when a fixed three-tier architecture with idle-time acceleration is available, and AS-AFL for large-scale decentralized networks requiring gossip-based peer-to-peer coordination without centralized synchronization points [142–144].

These frameworks represent a significant evolution beyond standard asynchronous protocols, embracing hybrid synchronization models that are intricately linked to the underlying network architecture. ASAFL [142] operates within a traditional client-server model but introduces a hybrid nature by making the degree of synchronicity a dynamically optimized parameter. EEFL [143] explicitly designs its protocol for a hierarchical architecture, using asynchronous communication at the lower level (client-to-edge) and synchronous communication at the upper level (edge-to-cloud). Pushing this further, AS-AFL [144] proposes a fully decentralized model where synchronous gossip protocols are used for fast collaboration within self-organizing client groups, while asynchronous updates are shared between them. A critical enabling technology across these diverse architectures is intelligent client grouping, though the strategic goal differs: ASAFL implicitly groups by speed, EEFL groups dissimilar clients to maximize local diversity, while AS-AFL groups similar clients into communities to foster efficient collaboration. This trend indicates that the choice of synchronization protocol is evolving into a complex, system-level design problem that is co-dependent on the network topology.

7.2.3 Scalable and Communication-Efficient Asynchronous FL

Gauthier et al. (2023) [145] introduce PAO-Fed, a communication-efficient asynchronous online federated learning (FL) algorithm that utilizes partial sharing of model parameters to address heterogeneous, time-varying client participation and delayed updates in distributed environments. The method targets a significant reduction of communication overhead while maintaining robust convergence under non-IID, imbalanced data distributions, and unpredictable client availability and communication delays. PAO-Fed achieves this by enabling clients and the server to exchange only dynamically selected subsets of model parameters by using evolving diagonal selection matrices, instead of full models, yielding up to 98% reduction in communication load. The algorithm projects data into a fixed-dimensional random Fourier feature (RFF) space for nonlinear regression, which is data-independent and resilient to model changes during online learning. The algorithm is designed for asynchronous federated learning by using a probabilistic model (Bernoulli process with changing probabilities) to represent when each client is available, which reflects real-world differences and unpredictability in client participation. PAO-Fed employs a delay-aware aggregation mechanism at the server, where the influence of each client update is attenuated according to its delay duration using a weight-decreasing function, thereby mitigating the negative impact of outdated information on the global model. Furthermore, clients autonomously perform local updates on their models when not participating in communication rounds, so their subsequent transmitted model portions are more refined and incorporate the latest local data. Theoretical analysis provides first- and second-order convergence guarantees and expressions for steady-state mean square deviation. The simulations, conducted on both synthetic and real-world datasets, demonstrate that PAO-Fed can achieve convergence properties comparable to standard online FL methods while reducing communication requirements by up to 98%. However, the current framework is centralized around a single server, which could become a bottleneck in applications with a very large number of clients.

Xie et al. (2024) [146] propose a decentralized federated learning framework with asynchronous parameter sharing, specifically to address communication inefficiency and the risk of model divergence in large-scale IoT networks. The authors create a decentralized federated learning system where devices work together in a peer-to-peer manner, without a central server, and share their model updates asynchronously to reduce communication delays. The proposed solution uses a decentralized federated learning approach in which each node updates its model using its own local data and aggregated parameters that are asynchronously shared by neighboring nodes. Nodes do not rely on a central server or wait for all others to finish before updating, so they can send and receive model updates independently. This asynchronous parameter sharing allows learning and communication to run in parallel, which makes the process more efficient and resilient to network delays. To make communication more reliable, the paper introduces a node scheduling strategy where nodes are allowed to exchange parameters only if the received signal-to-interference-plus-noise ratio (SINR) exceeds a predefined threshold, which ensures that only nodes with sufficiently high communication quality participate in parameter sharing. It also includes an optimal bandwidth allocation algorithm that distributes the available bandwidth among scheduled nodes to maximize the minimum data rate across all nodes, thereby minimizing the maximum transmission delay in the network. The paper's analysis shows that by reducing these communication delays, the learning process becomes quicker, along with faster model convergence, lower transmission delay, higher testing accuracy, compared to FedAv, AdaFL, and Semi-asynchronous FL model (FedSA), and more dependable for all nodes in the network. The paper theoretically shows, by applying the Shapley-Folkman lemma that the aggregated objective function becomes less non-convex with an increase in the number of nodes and improves the likelihood of global convergence. However, the theoretical results assume that data is uniformly distributed across nodes, which is often not the case in real-world IoT deployments.

The pursuit of scalable and efficient asynchronous FL, as highlighted by these works, is proceeding along two distinct architectural paths: radical optimization of the traditional centralized model and the adoption of fully decentralized, serverless paradigms. PAO-Fed [145] exemplifies the first path. It operates within a client-server topology but achieves massive communication efficiency by innovating on the content of the updates, exchanging only partial model parameters instead of the full model. This preserves the control of a centralized system while mitigating its primary bottleneck. In contrast, the framework by Xie et al. [146] follows the second path by getting rid of the central server entirely. Instead of relying on a coordinator, they make things efficient by managing the network itself, smartly scheduling which nodes talk when, and allocating bandwidth based on actual communication needs. This is a clear trade-off: the centralized way is easier to manage but can get choked by that one server; the decentralized way scales better but makes the whole system messier to coordinate. The future direction for massive-scale AFL will likely depend on the specific trade-offs each application can tolerate.

Table 6 summarizes recent asynchronous and semi asynchronous federated learning methods in terms of staleness handling, dataset, communication efficiency, and key results.

Table 6: Comparative summary of asynchronous and semi-asynchronous federated learning algorithms

Algorithm/ Reference	Staleness/Adaptivity handling	Datasets & scale	Communication & efficiency	Key outcomes/ Remarks
AsyFL/Asy-PPFL (Miao et al., 2024) [138]	Time-weighted stale model aggregation (SMA) and symmetric homomorphic encryption (SHE)	MNIST, Fashion-MNIST, CIFAR-10; up to 80% stragglers	10× lower overhead vs. CKKS; ~3× faster encryption	Sustains ~58% CIFAR-10 accuracy under 80% stragglers; privacy-preserving and convergence proven
Zhu et al. (2023) [139]	First-order Taylor staleness compensation with UCB-based client selection	Simulated wireless FL; variable latency clients	Latency-aware scheduling; no CSI required	Faster convergence, higher test accuracy under realistic wireless delays
Hu et al. (2023) [140]	Periodic aggregation with Age of Local Update (ALU) weighting	Wireless FL; heterogeneous clients	Aggregation period $\tilde{T} = T_{\max}/4$ optimal trade-off	Higher accuracy vs. FedAsync/FedAvg under limited bandwidth
FedLC/SC-FedLC (Xu et al., 2023) [141]	Local collaboration and adaptive learning rate for staleness compensation	CIFAR-10, EMNIST, Image-100; edge settings	Reduced network traffic (71.7 GB vs. 87.3 GB)	7% higher accuracy on CIFAR-10; improved non-IID generality
ASAFL (Chen et al., 2025) [142]	Dynamic synchronous degree via Lyapunov optimization; gradient calibration	MNIST, CIFAR-10	Adaptive per-round sync; higher per-round overhead	3.2%–7.8% accuracy gain; 1.6×–4.3× speedup to target accuracy

(Continued)

Table 6 (continued)

Algorithm/ Reference	Staleness/Adaptivity handling	Datasets & scale	Communication & efficiency	Key outcomes/ Remarks
EEFL (Sha et al., 2024) [143]	Semi-asynchronous client–edge sync with contrastive learning	MNIST, CIFAR-10, Fashion-MNIST	3-tier edge-cloud; SVD cosine clustering	2.5–7× faster convergence; 0.94 MNIST, 0.86 CIFAR-10 accuracy
AS-AFL (Zhou et al., 2024) [144]	Meta-learned segmentation; hybrid sync–async gossip aggregation	EMNIST; 1000+ nodes	90% lower WAN cost; 542 s to 93% accuracy	Scalable, decentralized; high accuracy (94%)
PAO-Fed (Gauthier et al., 2023) [145]	Partial parameter sharing with delay-aware weighting	Synthetic, real-world datasets	Up to 98% communication reduction	Converges comparably to full sharing; online FL capable
Decentralized Async FL (Xie et al., 2024) [146]	Peer-to-peer async parameter exchange with SINR-based scheduling	IoT networks; decentralized topology	Optimized bandwidth allocation; no central server	Faster convergence, reduced delay, improved reliability

8 Literature Reviews on Privacy and Security in Federated Learning

8.1 Data Leakage

8.1.1 Gradient Leakage Attacks and Analysis in FL: Models and Methods

Fang and Zhang (2024) [147] propose DLG-MIA, an enhanced DLG attack for an honest-but-curious attacker with extensive model knowledge and gradient access, that deepens privacy risks in federated learning by combining deep leakage from gradients (DLG) and membership inference attack (MIA) techniques. DLG reconstructs data from gradient updates, while MIA determines membership in the training set. The solution integrates MIA-derived membership probabilities into the DLG framework. The MIA model uses shadow training to mimic the target model and distinguish between members and non-members, improving confidence in reconstructed samples. The attack also uses gradient sign analysis to extract ground-truth labels, enabling selection of a class-specific MIA model. On CIFAR-10 with CNN, DLG-MIA reconstructs images 70 iterations faster than standard DLG (100 vs. 170 iterations): PSNR 23.48 vs. 21.80, SSIM 0.9085 vs. 0.8779, LPIPS 0.1011 vs. 0.1290 across 500–3000 images. Attack effectiveness decreases as the target model converges. However, no defense mechanism is evaluated, representing a critical vulnerability in FL systems where gradients thought to be safe are shown to leak sensitive training membership and visual data. The threat model assumes the attacker knows the model architecture and training; as models are more trained, attacks become harder and slower.

Yang et al. (2025) [148] propose a fast generation-based gradient leakage attack (FGLA) enabling an honest-but-curious server to reconstruct training data directly from highly compressed gradients in federated learning (FL). It uses a feature separation technique, Feature Separation From Gradients (FSG) that isolates the feature vector of each data feature from batch-averaged gradients of the fully connected layer by exploiting the fact that the gradient row corresponding to a data sample's true label is much larger. These extracted features are then input to a pre-trained generator, architecturally akin to an inverted ResNet, which efficiently reconstructs images without iterative refinement. On ImageNet, FGLA is at least 3000 times faster than optimization methods like DLG (e.g., 1.72 s vs. 13:59:47) with superior quality (PSNR 18.562 vs. 5.689), and robustly handles large batches (B = 256), high gradient compression (0.8% ratio), and Gaussian noise (0.1 variance). This generation-based approach trades generality for speed. Its primary weakness is a critical failure when duplicate labels appear in a batch, as it cannot distinguish their features and reconstructs a single, merged, false image.

Geng et al. (2024) [149] propose an improved gradient inversion attack for federated learning that works against both Federated Stochastic Gradient Descent (FedSGD) and FedAVG by introducing a zero-shot batch label restoration method that handles duplicate labels, overcoming prior assumptions that each batch contains unique labels and that attacks apply only to FedSGD protocols. This label method is highly effective; on ImageNet with 8 duplicates, it achieves 99.60% accuracy, while GradInversion fails at 49.39%. Their framework enhances image reconstruction via initialization, regularization (total variation, clipping, scaling), and group consistency techniques over multiple updates, which improves PSNR on ImageNet to 23.240 (from a 20.029 baseline). They extend attacks to FedAVG by approximating gradients from model weight differences across local epochs. Their method outperforms prior works on large-scale datasets like ImageNet, achieving a PSNR of 19.122 on duplicate-label batches, where GradInversion only gets 10.639. However, the attack has significant limitations. Scalability remains a challenge, as quality degrades with larger resolutions and batch sizes, with reconstructions becoming "blurred" and showing only outlines at batch size 128. Furthermore, their own method causes confused images when duplicates are present, and their FedAVG attack is only effective on small-scale data like CIFAR-10, failing on ImageNet.

Advanced gradient leakage research aims to overcome the ideal, lab-like conditions of early methods like DLG, targeting more realistic Federated Learning (FL) scenarios [147–149]. Geng et al. address the immaturity of existing attacks, which often fail against the widely-used FedAVG protocol or when batches contain duplicate labels [149]. Yang et al. tackle the critical performance bottlenecks of optimization-based attacks, namely their slow convergence (often hours per batch) and inability to handle the highly compressed gradients common in production FL [148]. Fang et al. seek to improve the fidelity and speed of DLG by fusing it with another known vulnerability, the Membership Inference Attack (MIA) [147]. These works introduce novel attack frameworks: Geng et al. provide a general optimization-based framework effective against both FedSGD and FedAVG, using an improved zero-shot label restoration method that successfully handles duplicate labels on datasets like CIFAR-10 and ImageNet [149]. Yang et al. pioneer a generation-based attack (FGLA), which uses a feature separation technique to extract individual sample features from the averaged gradient of the final fully connected layer [148]. A pre-trained generator then maps these features to images, achieving reconstruction in milliseconds (e.g., ≈1.7 s for a batch of 8, vs. ≈1.4 h for optimization methods) and succeeding with large batches (B = 256) and high compression (0.8% ratio) [148]. Fang et al. propose DLG-MIA, a hybrid optimization attack that adds a loss term to maximize the reconstructed image's MIA probability [147]. On CIFAR-10, this results in faster convergence (e.g., 100 vs. 170 iterations for a visible image) and superior quantitative metrics like PSNR and SSIM [147].

Despite these significant advances, the attacks operate under an honest-but-curious server threat model, assuming access to individual, de-noised client updates [147–149]. Their effectiveness is often contingent on

specific, fragile conditions. For instance, the generation-based FGLA requires a similar auxiliary dataset to pre-train its generator [148], and its core feature separation technique critically fails when duplicate labels appear in a batch. In such cases, it cannot distinguish the features and reconstructs a single, merged, false image for the duplicated entries [148]. Geng et al.'s method, while handling duplicates, also notes that the resulting images can appear confused, merging content from different images sharing the same label [149]. Furthermore, the DLG-MIA attack's potency degrades as the target model converges. It is highly effective at early training stages but struggles to reconstruct images from the more subtle gradients of a well-trained model (e.g., epoch 50) [147]. This evolution from slow optimization to fast generation [148] and hybrid [147] methods reveals a clear trade-off: FGLA achieves unprecedented speed and compression resistance but is brittle against duplicate labels [148]. This highlights that no single attack is universally dominant, and a pressing need for standardized benchmarks to evaluate attacks and defenses against a more complex and realistic set of FL conditions, such as data heterogeneity (non-i.i.d.), client sampling, and the cumulative effects of multi-round aggregation.

Table 7 provides a quantitative comparison of recent Gradient Leakage Attacks (2024–2025) in Federated Learning, summarizing attack methods, capabilities, example datasets, and key evaluation metrics.

Table 7: Quantitative comparison of gradient leakage attacks in FL (2024–2025)

Study (Attack)	Capabilities/Access	Example/Settings	Metrics/Notes
Fang et al., 2024 (DLG-MIA) [147]	Hybrid optimization (DLG+MIA); gradients & model known	CIFAR-10, CNN, 100 iters	PSNR 23.48 vs. 21.80; SSIM 0.9085 vs. 0.8779; LPIPS 0.1011 vs. 0.1290; most effective early training
Yang et al., 2025 (FGLA) [148]	Generation-based; gradients; pre-trained generator	ImageNet, B = 8	PSNR 18.562 vs. 5.689; Time 1.72 s vs. 13 h 59 m; fails on duplicate labels; robust to large batches
Geng et al., 2024 (Improved GI) [149]	Optimization-based; gradients; FedSGD/FedAVG; zero-shot label restoration	ImageNet, B = 16	Label Acc 99.6% vs. 49.4% (at Rep = 8); PSNR 19.12 vs. 10.64 (at Rep = 1); PSNR 23.24 vs. 20.03 (baseline); degrades at B ≥ 128; FedAVG fails large datasets

Note: HBC = honest-but-curious server (follows FL protocol but infers private data). PSNR = Peak Signal-to-Noise Ratio (higher better), SSIM = Structural Similarity Index (0–1, higher better), LPIPS = Learned Perceptual Image Patch Similarity (lower better), Time = reconstruction per batch (lower better), Label Acc = label accuracy (higher better). Comparison values ("vs.") typically refer to the proposed method versus the relevant baseline (e.g., standard DLG or GradInversion).

8.1.2 Defense Mechanisms Against Gradient Leakage in FL

Structural and Foundational Defense Approaches

Liu et al. (2023) [127] propose Privacy-Encoded Federated Learning (PEFL) to defend against honest-but-curious adversaries conducting gradient-based data reconstruction attacks (DLG, iDLG, IG). PEFL decomposes each weight matrix into three cascading sub-matrices: a column representation, a central matrix computed via nonlinear transformation, and a row representation. Only the column and row sub-matrices

are transmitted, introducing an entangling nonlinear mapping between the gradients and raw data. This method breaks the classical defense trade-off of sacrificing accuracy for security. The method evaluated on MNIST/LeNet and CIFAR-10/VGG, PEFL achieves a 20dB PSNR reduction compared to DP and GC defenses, rendering images unrecognizable. It also thwarts label extraction, reducing iDLG success to 62.90%. PEFL maintains or slightly improves model accuracy (1.2% gain) while reducing inference computational complexity by 2.7× and communication cost by 2.1×. The primary trade-off is that its security depends on keeping the sampling indexes private.

Hu et al. (2024) [128] investigate whether differential privacy (DP) defends federated learning against gradient leakage attacks from honest-but-curious adversaries, considering the privacy-utility trade-off. Evaluated on MNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet with LeNet, ResNet-18/34, and VGG-11, Central Differential Privacy (CDP) with per-layer clipping and Local Differential Privacy (LDP) with moderate noise ($\epsilon \approx 10$) defend against existing GLAs in shallow models. However, both significantly degrade utility in deeper architectures, with ResNet-18 accuracy dropping 41.3% (LClip) and 43.8% (FClip) on CIFAR-10 under LDP. Per-layer clipping outperforms flat clipping by altering gradient direction rather than just magnitude. The paper proposes an improved GLA integrating clipping norm estimation into the optimization objective, which neutralizes clipping protection and completely recovers data from CDP-protected gradients. The study focuses only on optimization-based GLAs, not analytics-based attacks.

These papers investigate two distinct foundational strategies for defending against gradient leakage: fundamentally re-engineering the model architecture (Liu et al. [127]) and critically evaluating the most common defense, Differential Privacy (DP) (Hu et al. [128]). Liu et al. propose Privacy-Encoded Federated Learning (PEFL), a novel structural defense that decomposes weight matrices and uses a nonlinear transformation [127]. Hu et al. analyze whether DP actually works, questioning its high utility cost and fundamental security [128].

The core tension between these papers lies in the classic privacy-utility trade-off. Liu et al. claim their PEFL method breaks this trade-off, achieving a 20dB PSNR reduction (making images unrecognizable) while simultaneously improving model accuracy by 1.2% and increasing efficiency [127]. This directly counters the findings of Hu et al., who demonstrate that this trade-off is severe for DP [128]. Their work shows that while DP can protect shallow models, it causes a catastrophic drop in utility for deeper architectures, with ResNet-18 accuracy falling by over 41% [128]. Furthermore, Hu et al. demonstrate that DP's protection is ultimately brittle. They propose an improved attack that estimates the defense's clipping norm, allowing it to neutralize the protection and completely recover data [128]. This suggests that defenses based on simple gradient manipulation, like DP, are fundamentally vulnerable to adaptive attackers [128].

Adaptive and Dynamic Defense Mechanisms

Wang et al. (2024) [94] investigate gradient leakage attacks from honest-but-curious servers in production federated learning (FL) and propose an adaptive defense mechanism, OUTPOST. Prior works overestimate attack impact through impractical assumptions like direct gradient sharing, explicit initialization, and single-step training. Real FL naturally mitigates risks via multi-step training, non-IID data, and standard initialization. Evaluated on EMNIST and CIFAR-10 with LeNet, existing attacks achieve limited reconstruction quality. OUTPOST selectively perturbs gradients with Gaussian noise per layer based on privacy risk measured by weight variance and Fisher information, adapting noise over iterations. It achieves 3.28% and 2.19% accuracy loss with minimal overhead (3.54 and 1.47 s delay) on EMNIST and CIFAR-10, respectively. While OUTPOST is not optimal for all privacy metrics (MSE, SSIM, LPIPS), it balances accuracy, overhead, and protection better than differential privacy, gradient compression, Soteria, and GradDefense.

Chang and Zhu (2024) [129] propose two gradient-based defense methods against data leakage in vertical federated learning. The work models a Byzantine threat, defining the attacker as a curious and malicious server who seeks to infer client data from shared gradients. The Pseudo-Gradient Defense Method (Method 1) combines current and previous gradients using cosine similarity-weighted aggregation to obscure updates. The Gradient Sparsification Defense Method (Method 2) adaptively retains only significant gradients based on average cosine similarity across clients. Experiments on MNIST and Cifar-10 datasets (800 images, batch size 40) demonstrate effective privacy protection. Method 1 achieves minimum PSNR values of 11.58 (MNIST at Grt = 0.2) to 11.56 (Cifar-10 at Grt = 0.2). Method 2 reaches minimum PSNR values ranging from 19.91 (MNIST) to 10.52 (Cifar-10), depending on sparsification percentage (Sp = 87.5%). Both methods maintain global model accuracy comparable to baseline FedAvg. However, extreme sparsification (Sp = 87.5%) degrades accuracy by a maximum difference of about 15% compared to normal training on Cifar-10, revealing a critical privacy-utility trade-off. The methods do not explicitly evaluate against adaptive attackers aware of these defenses.

Hu et al. (2024) [95] propose an Adaptive Privacy-Preserving Federated Learning (Adp-PPFL) framework to defend against Gradient Leakage Attacks (GLA) under an honest-but-curious server model. A server-side leakage risk-aware privacy decomposition mechanism dynamically allocates privacy budgets based on quantified leakage risks per round and client. A client-side adaptive privacy-preserving local training mechanism adaptively adjusts noise and clipping during local training, intentionally adding larger noise in early, high-risk stages to strengthen the defense. Experiments on MNIST, CIFAR-10, and Labeled Faces in the Wild (LFW) show Adp-PPFL effectively prevents data reconstruction. For example, in the early stage (R1-R5) on MNIST, Adp-PPFL achieves a reconstructed PSNR of 7.44 (indicating high protection), whereas the non-private FL baseline yields a high PSNR of 66.56 (indicating high leakage). This robust initial privacy introduces a trade-off, as it comes at the cost of lower model accuracy in early training rounds. However, the framework ultimately achieves better model convergence, reaching 93% accuracy within 50 rounds on MNIST, which outperforms the Fixed-PPFL baseline's final accuracy of 92% in the same timeframe.

These studies pivot from static, high-cost defenses to dynamic mechanisms that adapt protection based on immediate risk. This approach challenges the idea that all gradients pose an equal threat at all times. Both Wang et al. (2024) [94] and Hu et al. (2024) [95] develop adaptive noise-based defenses for horizontal FL but use different risk assessments. Wang et al. argue that production FL is already robust, so their OUTPOST defense is lightweight. It applies minimal, selective noise based on internal model metrics like the Fisher information matrix [94]. In contrast, Hu et al. quantify risk at the round and client level. Their Adp-PPFL framework uses a more aggressive strategy by intentionally adding larger noise in high-risk early rounds [95]. Chang and Zhu adapt this philosophy to Vertical FL with two novel, non-noise methods. These methods use cosine similarity to either obscure gradients by combining updates or to adaptively sparsify them [129].

This shift to adaptive strategies reveals a complex landscape of privacy-utility trade-offs. The OUTPOST defense provides a balanced trade-off, achieving solid protection with a negligible accuracy drop (e.g., 2.19% on CIFAR-10) and minimal overhead [94]. Adp-PPFL introduces a temporal trade-off. It deliberately sacrifices accuracy in the early rounds for strong privacy but ultimately achieves better final model convergence than a fixed-noise baseline [95]. The methods from Chang and Zhu show a more critical trade-off. While their pseudo-gradient and moderate sparsification methods maintain accuracy, extreme sparsification (87.5%) causes the defense to fail. This induces a severe accuracy loss of about 15% on Cifar-10, highlighting a breaking point where the privacy-utility balance collapses [129]. Furthermore, the defenses target different threat models. Wang et al. and Hu et al. assume an honest-but-curious server [94,95], while Chang and Zhu target

a stronger Byzantine malicious server [129]. A key limitation is that these mechanisms are not evaluated against adaptive attackers who might be aware of the defense and alter their strategy.

Table 8 shows the summary of Gradient Leakage Attacks (GLAs) in Federated Learning, including key studies, attacker capabilities, example datasets, model settings, and evaluation metrics.

Table 8: Key gradient leakage attacks in federated learning with capabilities and metrics

Study (Attack)	Capabilities/Access	Example/Settings	Metrics/Notes
Hu et al., 2024 [128]	Transformation-based; estimates clipping norm to break DP defenses. Access: Gradients/Updates (HBC).	MNIST, CIFAR, GTSRB; ResNet-18/34, VGG.	Breaks CDP/LDP[f]; proves FClip insufficient against GLAs.
Fang et al., 2024 (DLG-MIA) [147]	Hybrid; integrates MIA loss constraints; extracts label via gradient signs. Access: Gradients, MIA model (HBC).	CIFAR-10; Standard CNN.	Higher PSNR[b]/SSIM[c] than DLG; faster convergence.
Yang et al., 2025 (FGLA) [148]	Generation-based; feature separation from FC layer gradients. Access: Gradients, feature extractor (HBC[a]).	ImageNet, CelebA; $B \leq 512$; ResNet-50, EfficientNet.	Fast (ms); robust to high compression (0.8%); no optimization needed.
Geng et al., 2024 (gDLG) [149]	Attacks FedAVG (weights) & FedSGD; One-shot batch label restoration. Access: Gradients/Weights (HBC).	ImageNet, CIFAR-10; $B \in [16, 128]$; LeNet.	Uses group consistency; handles duplicate labels; uses Image Alignment.

Note: [a] **HBC:** Honest-but-curious server. [b] **PSNR:** Peak Signal-to-Noise Ratio (↑ better). [c] **SSIM:** Structural Similarity Index (↑ better). **LPIPS:** Learned Perceptual Image Patch Similarity (↓ better). [f] **CDP/LDP:** Central/Local Differential Privacy.

8.1.3 Privacy-Preserving Architectures and Secure Encoding Techniques

Madni et al. (2023) [150] propose a blockchain-based Swarm Learning (SL) framework to mitigate gradient leakage from honest-but-curious attackers in FL. Unlike FL which shares perturbed gradients with a central server, SL shares original gradients only among blockchain-authenticated nodes through smart contracts. Gradient leakage attacks (DLG, IGA, GGL using GANs or Bayesian/CMAES optimization) can reconstruct original training data from shared gradients. SL eliminates this attack vector by decentralizing aggregation to a randomly selected sentinel node using FedAvg, with blockchain securing node authentication. On CIFAR-10 and MNIST with ResNet18 and CNN-2 models across non-IID distributions (Dirichlet α: 0.1–10), SL achieves higher accuracy than individual node training (α = 10: MNIST ResNet18 98.88% ± 0.18% vs. baselines with 2%–10% accuracy loss from perturbation). The approach maintains baseline model performance without noise degradation required by differential privacy or sparsification defenses, while preventing gradient reconstruction attacks. However, blockchain overhead and sentinel node selection add computational latency, and scalability with numerous nodes remains unexplored.

Chen et al. (2024) [151] propose a Compressed Sensing (CS)-based privacy-preserving FL scheme protecting against gradient leakage attacks from honest-but-curious servers and edge nodes. Gradient leakage attacks (DLG, iDLG, IG, GI, LLG using gradient matching) can reconstruct images and extract labels from shared gradients. The scheme compresses gradients using CS measurement matrices and projects them onto nonnegative space, eliminating negative values that reveal label information. Double aggregation (local then global) prevents individual gradient disclosure. On MNIST and CIFAR-100 with LeNet models (B = 1, B = 4 batches), the method achieves ≤1% accuracy loss compared to plain FL while achieving SSIM ≤ 0.15 vs. 0.8+ undefended, MSE > 140 vs. ≈0 undefended, and label extraction accuracy ≈10% (LLG attack) vs. 95%+ baseline. Communication reduces 30%–70% (compression ratio c: 0.3–0.9); computation overhead is minimal (matrix multiplication only). Against advanced attacks with known labels, reconstructed images remain highly noisy. Compared to DP (50% accuracy loss at severe perturbation), GC (2% loss), Soteria (ineffective vs. iDLG), and OUTPOST (2% loss), the CS-based method balances privacy and utility. Sparsity-adaptive gradient reconstruction (SAAIHT) adaptively estimates true gradient sparsity post-aggregation for improved decompression accuracy.

Chen et al. (2024) [152] address privacy vulnerabilities from gradient inversion attacks launched by malicious servers in federated learning by proposing an autoencoder-based parameter compression method. Each client uses a lightweight 1D convolutional autoencoder to compress and perturb local model parameters, sharing only low-dimensional representations for aggregation. Empirical results on Fashion-MNIST, CIFAR-10, and CIFAR-100 with LeNet and MLP show a 4.1× reduction in communication costs compared to FedAvg and comparable accuracy to unprotected training. The paper reports that its proposed method resulted in only marginal accuracy drops (e.g., 0.32%–1.50% on CIFAR-10 and 0.51%–1.14% on non-IID Fashion-MNIST), unlike the significant degradation observed with high sparsification or noisy gradients. The approach effectively blurs reconstructions, which prevents successful gradient inversion attacks across IID and non-IID splits.

Instead of relying on traditional noise-based defenses for privacy, these studies propose two different solutions: 1) fundamentally changing the system's design (structural decentralization) and 2) using advanced compression techniques on the model's updates (parameter encoding). Madni et al. (2024) [150] propose the most significant architectural shift by eliminating the central server. Their Swarm Learning (SL) framework uses a blockchain for node authentication, allowing a randomly selected sentinel node to securely aggregate the original, unperturbed gradients [150]. In contrast, both Chen et al. 2024 [151] and Chen et al. (2024) [152] papers retain a central server but focus on compressing the parameters to simultaneously reduce communication and obscure private data. Chen et al. use Compressed Sensing (CS) to compress gradients and, critically, project them onto a non-negative space to explicitly block label extraction attacks that rely on negative gradient values [151]. Chen et al. employ a learned defense, using a 1D convolutional autoencoder on each client to compress parameters into a low-dimensional representation, where the lossy compression itself acts as the privacy-preserving perturbation [152]. The primary motivation for these approaches is to break the severe privacy-utility trade-off inherent in defenses like Differential Privacy. Madni et al.'s Swarm Learning achieves this by design, maintaining baseline model performance by using original gradients. However, it introduces a new trade-off of computational latency and potential scalability issues due to its blockchain overhead [150]. The encoding strategies from Chen et al. and Chen et al. demonstrate a more balanced and practical trade-off. They achieve both high privacy (SSIM ≤ 0.15 for the CS method) and significant communication reduction (4.1× for the AE method) with only marginal accuracy drops (e.g., <1.5%) [151,152]. A key distinction lies in their secondary benefits; the CS-based approach is one of the few to explicitly target label leakage, while the autoencoder method provides an adaptive perturbation inherently tied to the model parameters [151,152].

Table 9 summarizes key structural and compression-based privacy defenses in federated learning.

Table 9: Summary of structural and compression-based privacy defenses in federated learning

Study (Defense)	Capabilities/Access	Example/Settings	Metrics/Notes
Madni et al., 2023 [150] **(Swarm Learning)**	**Method:** Decentralized aggregation via blockchain-authenticated sentinel node. Eliminates central server. **Access:** Original gradients shared among authenticated nodes. **Threat Model:** HBC attackers.	**Dataset:** MNIST, CIFAR-10. **Model:** ResNet-18, CNN-2. **Non-IID:** Dirichlet α 0.1–10.	**Metrics:** Maintains baseline accuracy (e.g., MNIST ResNet18 98.88% ± 0.18%). **Notes:** Prevents gradient reconstruction (DLG, IGA, GGL). Trade-offs: blockchain latency, scalability unexplored.
Chen et al., 2024 [151] **(CS-Gradient Compression)**	**Method:** Compressed sensing (CS) projection; double aggregation; eliminates negative gradient values to prevent label extraction. **Access:** Gradients shared with central server. **Threat Model:** HBC server/edge nodes.	**Dataset:** MNIST, CIFAR-100. **Model:** LeNet, B = 1–4. **Settings:** Compression ratio 0.3–0.9.	**Metrics:** ≤1% accuracy loss; SSIM ≤ 0.15 vs. undefended 0.8+; label extraction ≈ 10%. **Notes:** Explicitly blocks label leakage; minimal computation; balances privacy-utility.
Chen et al., 2024 [152] **(Autoencoder Compression)**	**Method:** Client-side 1D convolutional autoencoder compresses and perturbs parameters; low-dimensional aggregation. **Access:** Compressed parameters shared with central server. **Threat Model:** Malicious server.	**Dataset:** Fashion-MNIST, CIFAR-10, CIFAR-100. **Model:** LeNet, MLP. **Batch Size:** variable.	**Metrics:** 4.1× communication reduction; marginal accuracy drop 0.32%–1.5%. **Notes:** Blurs reconstructions to prevent inversion; adaptive to IID/non-IID splits; practical trade-off.

8.2 Poisoning Attacks

8.2.1 Backdoor Attack Strategies

Xiao et al. (2024) [153] propose SBPA, a Byzantine attack targeting federated learning systems with honest servers, especially in AIoT-FL networks where large amounts of distributed data are generated in real-time. Malicious participants inject backdoor triggers into portions of source class images and relabel them as target classes, while creating sybil nodes to exploit network instability and increase the aggregation probability of their poisoned local models. The method maintains high classification accuracy on non-backdoor samples while achieving misclassification for backdoor images, making detection difficult. Experiments on F-MNIST, CIFAR-10, and FEMNIST under both IID and non-IID data distributions demonstrate that SBPA outperforms state-of-the-art methods across multiple metrics. Key experimental trade-offs show: as

malicious participants increase from 10% to 40%, SBPA achieves very high backdoor accuracy (BDAcc, over 90%), while main task accuracy (MTAcc) degrades minimally (<5%); with 5 sybil nodes per attacker and a 0.4 poisoning rate, SBPA achieves robust performance but requires sufficient malicious participation for practical effectiveness.

Mbow et al. (2024) [154] present a Byzantine-style data poisoning attack on federated learning-based Network Intrusion Detection System (NIDS), targeting Mirai traffic misclassification as benign in a multiclass setting. Using a conditional Wasserstein GAN with gradient penalty (WGAN-GP), the attack generates realistic poisoned Mirai samples labeled as benign. The approach applies SHAP to constrain perturbations to non-functional features, preserving attack functionality and stealthiness. Evaluated on the CIC-IoT-2023 dataset with 10 IID clients, the global model trains over 10 rounds. Results show a one-time poisoning attack degrades accuracy modestly and requires multiple malicious clients for effect (e.g., with 4 clients, Mirai detection drops from 99.67% to 76.57%). Gradual poisoning proves more effective, with a single malicious client reducing accuracy from 83.01% to 57.25%, Mirai detection rate from 99.67% to 91.03%, and increasing false negatives. The study highlights the trade-off between attack effectiveness and stealth, and the need for defense mechanisms in FL-NIDS. Limitations include a focus on Mirai only and the IID data assumption.

Zhou et al. (2025) [155] introduce a meta-reinforcement learning-based framework for Byzantine untargeted model poisoning attacks against federated learning systems, effective even under robust aggregation defenses. The attack infers an approximate global data distribution using a Conditional Generative Adversarial Network (CGAN) trained on malicious clients' data, enabling local simulation of FL environments. Using meta-reinforcement learning, the attack trains a policy adaptable to various aggregation strategies without prior knowledge of the actual one. This framework represents a shift from heuristic methods by employing a "scaling and noise injection attack". This precision-focused technique applies a unique, meta-learned policy to each layer's gradient, unlike uniform modifications. Its demonstrated efficacy is high, reducing model accuracy to approximately 10% across robust defenses on standard benchmarks. However, this generalization introduces a significant trade-off: the meta-learning phase requires high computational overhead to simulate diverse aggregation strategies. This class of attack also exhibits a strong dependency on attacker density, diminishing significantly with 15% or fewer malicious clients. While its ability to defeat unseen defenses like Bulyan highlights a potent new threat vector, its scalability and adaptation to targeted attacks remain open research questions.

Poisoning attack research is rapidly evolving from simple data manipulation to sophisticated, adaptive strategies. This evolution is captured by three distinct approaches. Xiao et al. (2024) [153] introduce an amplification attack, SBPA, which exploits the network instability of AIoT environments. By creating sybil nodes, the attacker increases the aggregation probability of their backdoor, while achieving high attack success with minimal impact on the main task. Mbow et al. (2024) [154] focus on stealth for FL-based Network Intrusion Detection Systems (NIDS). It uses a conditional WGAN-GP to generate poisoned Mirai attack data that mimics benign traffic. Their use of SHAP to modify only non-functional features ensures the attack remains both functional and evasive. Zhou et al. (2025) [155] present a highly generalizable attack, using meta-reinforcement learning to train a policy that defeats robust aggregation rules without prior knowledge. This Meta-Reinforcement Learning (MRL)-based framework learns to apply a unique scaling and noise injection attack to each model layer, resulting in a devastating attack that drops model accuracy to around 10% [155].

These advanced methods reveal new and critical trade-offs. The MRL attack (Zhou et al.) achieves unparalleled effectiveness and generalization, but this power comes at the cost of extremely high computational overhead for its local simulation and meta-learning phase [155]. Furthermore, both the MRL attack

and SBPA (Xiao et al.) show a strong dependency on the number of malicious participants, with their effectiveness diminishing significantly when attackers control a small fraction of the network (e.g., under 15%) [153,155]. The stealthy GAN attack (Mbow et al.) highlights a temporal trade-off, demonstrating that a "gradual" poisoning strategy is far more effective than a "one-time" injection, which the global model eventually overcomes [154]. While powerful, these attacks are specialized: Mbow et al. focus on IID data for NIDS [154], while Xiao et al. target the unique vulnerabilities of AIoT networks [153].

Table 10 presents a comparative summary of backdoor and poisoning attack strategies in federated learning, describing attack methods, experimental setups, and key performance metrics.

Table 10: Comparison of backdoor and poisoning attacks in federated learning

Study (Attack)	Method/Capabilities	Experimental setup	Metrics/Key outcomes
Xiao et al., 2024 [153] (SBPA)	Byzantine amplification attack with sybil nodes; partial image triggers and relabeling; exploits AIoT-FL instability; honest server.	F-MNIST, CIFAR-10, FEMNIST; non-IID/i.i.d.; 5 sybil nodes/attacker; poisoning rate 0.4.	BDAcc > 90%, MTAcc ↓ < 5%; effective under 10–40% malicious clients; stealthy but attacker-density dependent.
Mbow et al., 2024 [154] (WGAN-GP Poisoning)	Byzantine data poisoning on FL-based NIDS; uses conditional WGAN-GP + SHAP to alter non-functional Mirai features; maintains attack functionality.	CIC-IoT-2023; 10 IID clients; multiclass NIDS; 10 rounds training.	Gradual poisoning: accuracy ↓ 83.01→57.25%, detection ↓ 99.67→91.03%; stealth ↑; limited to IID and Mirai.
Zhou et al., 2025 [155] (MRL-Poisoning)	Meta-RL-based untargeted model poisoning; CGAN infers global data; adaptive policy defeats robust aggregators (Bulyan, Trimmed Mean).	MNIST, FMNIST, CIFAR-10; 100 clients; various aggregation rules; malicious fraction 5%–20%.	Model accuracy ↓ ≈ 10% under robust defenses; generalizable but high computation and low scalability < 15% malicious.

Note: BDAcc: Backdoor Accuracy; MTAcc: Main Task Accuracy; ↓: decrease; ↑: increase. All attacks assume Byzantine clients. Metrics illustrate trade-offs between stealth, scale, and computational overhead.

8.2.2 Property and Data Poisoning Attacks

Wang et al. (2023) [13] introduce a Byzantine poisoning-assisted property inference attack (Poisoning-Assisted Property Inference—PAPI-attack), designed to infer unrelated data properties in dynamic FL settings. The attack's core contribution is linking poisoning to inference: a malicious participant injects property-specific poisoned data via label-flipping to distort the decision boundary. This amplifies the difference in model updates when the property is present, allowing a binary attack model (trained via shadow training on auxiliary data) to successfully infer the property. The method evaluated on CIFAR-10 and LFW, the attack achieves high Area Under the Curve (AUC) scores (often >0.8–0.9) even against converged models, where non-poisoning attacks (PI-attack) fail (AUC 0.5). This highlights a critical vulnerability that poisoned

models can be induced to leak more information. However, the attack shows key trade-offs and limitations: its effectiveness degrades as the number of participants increases, and it relies on a powerful attacker with full local control and auxiliary data.

Nowroozi et al. (2024) [156] explore Byzantine data poisoning attacks in FL-based NIDS (CIC and UNSW datasets) in a 2-client, white-box setting. The work highlights a fundamental asymmetry between attack vectors: Label Flipping (LF) is shown to be ineffective for stealth, as its high Attack Success Rate (ASR 95%) is coupled with a catastrophic drop in server accuracy (e.g., 96.8% to 4.28%), rendering it trivial to detect. In contrast, Feature Poisoning (FP), targeting high-importance features identified by Random Forest, demonstrates significant stealth. It successfully compromises the model (e.g., 1% FP on CIC yields 96.28% ASR) while maintaining high server accuracy (96.42%), making the attack undetectable by performance monitoring alone. The study finds FP's effectiveness varies non-linearly with the poisoning percentage. While limited by the white-box, 2-client setup and lack of defense evaluation, this work underscores that FP, not LF, is the more potent threat vector for network traffic data.

Thein et al. (2024) [157] propose pFL-IDS, a personalized federated learning-based intrusion detection system addressing non-IID data heterogeneity and Byzantine poisoning attacks on IoT networks. The method combines two main components: a personalized local model and a server-side poisoned client detector. Personalization is achieved using 1D-CNNs with a mini-batch logit adjustment loss to handle class imbalance from non-IID data. The server-side detector performs a two-step client similarity alignment using cosine similarity to identify and re-weight poisoned clients, crucially without requiring clean server data. Evaluated on the N-BaIoT dataset with 20 clients under IID and two non-IID scenarios (Dirichlet $\eta = 0.1$ and severe class imbalance), pFL-IDS demonstrated robustness. Against a 30% malicious client ratio across five poisoning attack types, it maintained high performance (e.g., >95% accuracy and F1-score in the challenging non-IID scenario 1) while keeping the Attack Success Rate (ASR) low (e.g., <7.5% in scenario 1 and <1.6% in scenario 2). In contrast, baseline methods like FedAvg, coordinate-wise median, and multi-Krum collapsed under these non-IID conditions, with ASRs exceeding 50%. The evaluation is limited to a 20-client, full-participation setting, and the authors note that large-scale IoT deployment would require client selection strategies. This work addresses a key gap in robust, personalized FL for heterogeneous networks.

The study of poisoning attacks reveals a critical divide between brute-force and stealthy methods. Nowroozi et al. [156] provide a clear example of this division. They demonstrate that simple label-flipping attacks against network intrusion systems are ineffective because they cause a catastrophic drop in server accuracy, making the attack obvious. However, their work proves that a more sophisticated Feature Poisoning (FP) attack, which targets only high-importance features, is highly effective. The FP attack successfully compromises the model with a 96.28% Attack Success Rate (ASR) while remaining stealthy by maintaining a high 96.42% server accuracy [156]. Wang et al. [13] showcase an even more advanced, hybrid threat. They propose a Poisoning-Assisted Property Inference (PAPI) attack, where the poisoning itself is not the goal. Instead, the attacker uses label-flipping to intentionally distort the model's decision boundary, which amplifies the signal of a secondary property. This allows an inference attack to succeed with a high AUC (over 0.9) even on a converged model, a scenario where normal inference fails [13].

This escalation in attack sophistication necessitates robust defenses that can operate under challenging, realistic conditions. Thein et al. (2024) [157] directly address this need for FL-based NIDS in non-IID environments. Their pFL-IDS framework introduces a dual defense. It first handles the data heterogeneity using a personalized logit adjustment loss, which prevents non-IID data from being mistaken for an attack. It then deploys a novel server-side detector that uses a two-step cosine similarity alignment to identify and remove malicious clients without needing a clean dataset. This defense proved highly effective. It maintained over 95% accuracy with a 30% malicious client ratio in a non-IID setting, a scenario where standard robust

aggregators like Median and Multi-Krum collapsed [157]. Together, these papers illustrate a clear arms race. While attackers (Wang et al. [13], Nowroozi et al. [156]) develop stealthy and hybrid poisoning methods, defenders (Thein et al. [157]) are moving toward adaptive, personalized models to counteract them.

Table 11 summarizes property- and data-poisoning attacks in federated learning, outlining each attack's capabilities, access type, empirical success rates, leakage impacts, defense overhead, and example trade-offs.

Table 11: Overview of property and data-poisoning attacks in federated learning

Attack type	Capability	Access	Success (ASR/AUC)	Leakage or Impact	Defense Cost	Example (Parameters & Trade-Offs)
Label-Flipping (LF) [156]	Mislabels local data to corrupt global updates	Full control of local labels	ASR ≈ 95%; accuracy ↓ to 4.28% (at 1% poison)	Model collapse; highly visible/suspicious	Not quantified; defenses fail on non-IID	p = 0.01 full flip. Higher p (e.g., 2%) keeps accuracy collapsed at ≈5.37%.
Feature Poisoning (FP) [156]	Perturbs key features via Random Forest selection	White-box to gradients	ASR = 96.28%; accuracy = 96.42% (CIC)	Target drift; low visibility (Ghost attack)	Not quantified (lack of defense discussion)	p = 0.01, value replacement via normalization. High ASR maintained even at low p.
PAPI (Poisoning-Assisted Property Inference) [13]	Adds poisons to distort boundaries and expose properties	Local control + shadow data	AUC > 0.90 (PAPI); AUC > 0.80 (Scaling)	Secondary property leakage (severe)	Pruning/ compression defenses are bypassed	Poison ratio = 50% (shadow), Scaling $\delta \geq 1$. Scaling resists aggregation but stability varies.
Byzantine Poisoning [157]	Sends arbitrary/scaled gradients to disrupt aggregation	Malicious clients (no data access)	ASR up to 100%; mitigated <8% via pFL-IDS	Model divergence, instability	Client selection needed for large-scale support	30% attackers; pFL-IDS cosine filter keeps ~96%–98% accuracy.

8.2.3 Backdoor and Poisoning Attack Defenses

Chen et al. (2024) [158] propose FLSAD, a defense against stealthy Byzantine backdoor attacks in federated learning. The method employs a two-stage process: recovering triggers via an entropy maximization estimator and using self-attention distillation to eliminate the backdoor. This defense does not require clean

data and leverages shallow, benign model layers to correct deep, malicious layers. On MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100, the experimental results show FLSAD consistently reduced the Attack Success Rate (ASR) to approximately 2% or less. This performance significantly outperformed all four baselines, including Robust Learning Rate (RLR) [159] and CONTRA [160]. Critically, main task accuracy (ACC) was maintained, for example, at over 93.4% on MNIST and over 86.8% on CIFAR-10. However, a key trade-off is its computational cost, as FLSAD is more expensive than FoolsGold but less costly than RLR. Its effectiveness also slightly diminishes as the trigger size increases.

To address the limitations of single-server models that rely on strong assumptions, Miao et al. (2024) [161] propose RFed. This defense against Byzantine poisoning attacks operates within a dual-server architecture with semi-honest servers. The core defense mechanism is a scaled dot-product attention module. This module assesses the similarity between client gradients and a trusted root gradient benchmark, systematically assigning lower softmax weights to updates that are dissimilar and therefore likely malicious. Using CKKS homomorphic encryption and obfuscation, RFed ensures neither server accesses plaintext updates. On MNIST, Fashion-MNIST, and CIFAR-10 with label-flipping and backdoor attacks, RFed achieves a poisoning attack failure rate higher than 96% (e.g., label-flip ASR ≤ 0.04 on MNIST; backdoor ASR ≤ 0.04 on CIFAR-10) while maintaining high accuracy. The convergence rate is proven to be $O(1/T^2)$. Critically, its computational complexity for weight calculation is $O(nd)$, which is linear and more scalable than the quadratic complexity of methods like Krum. Key trade-offs are the increased communication overhead from the dual-server architecture and a robustness guarantee limited to a malicious client ratio of less than 50.

As poisoning attacks grow more complex, defensive strategies have also evolved, moving beyond simple statistical filtering to more sophisticated, model-aware approaches. The two papers, although both focus on poisoning, take very different approaches. FLSAD by Chen et al. [158] uses a detect-and-repair method. It works after a backdoor has already affected the model and aims to remove its impact carefully using a self-attention distillation technique. This method takes advantage of the model's own internal structure to correct itself. In contrast, RFed by Miao et al. [161] follows a prevent and filter philosophy. It is a robust aggregation scheme that works proactively to minimize the impact of malicious updates during the training process by using an attention-based weighting mechanism. These approaches are highly complementary and could form a powerful layered defense: RFed could serve as the primary defense during aggregation, while FLSAD could be used periodically for a deeper "audit" and "cleansing" of the global model. A key trade-off lies in their prerequisites and costs: FLSAD is computationally intensive but notably does not require a clean dataset for its operation, whereas RFed's security relies on the availability of a small, trusted "root" dataset to generate its benchmark gradient, and its privacy guarantees come with the communication overhead of a dual-server, homomorphic encryption architecture.

Li et al.'s [162] privacy-preserving federated learning scheme (PFLS) defends against Byzantine (poisoning) attacks from malicious participants while protecting participant privacy from honest-but-curious servers using a dynamic adaptive defense mechanism based on Euclidean distance between participant gradients and a masked base gradient. The scheme employs multidimensional homomorphic encryption with hierarchical aggregation and identity-based aggregating signatures for data integrity, eliminating the need for a trusted third party. Experimental results on MNIST and CIFAR-10 datasets demonstrate detection rates of 100% for below 35 malicious participants (with an MLP model) and below 25 malicious participants (with a CNN model) on the non-IID MNIST dataset, and effective mitigation of poisoning attacks while maintaining classification accuracy (0.5%–15.3% improvement over undefended baselines). However, the approach incurs linear computational overhead growth with participant numbers and additional communication frequency of $2N + 2n$ per training round, requiring further parameter quantification optimization to improve transmission efficiency.

Shen et al. (2024) [163] propose SPEFL, an efficient framework for IoT. It is designed to be robust against poisoning attacks, a type of Byzantine threat where malicious devices submit crafted gradients, while also preserving privacy within a semi-honest server model. To mitigate high computational overhead, it uses a three-server architecture (two service providers, one assistant) to offload work from resource-limited devices. Using additive secret sharing, it performs robust aggregation on encrypted shares by weighting gradients based on their Pearson correlation to a securely computed median. Empirical validation on HAR, MNIST, and CIFAR-10 confirms high robustness against label-flipping, backdoor, and local model attacks, even with up to 50% malicious participants. This security is achieved with notable efficiency: its O (n) complexity yields execution speeds 141x faster than homomorphic encryption-based methods and 25.5x faster than general multiparty computation (MPC) approaches, critically reducing the device-side burden to simple additions. The authors also propose a security-enhanced verifiable protocol to address malicious servers (a stronger Byzantine model) with a 23% runtime overhead. A key limitation, however, remains the evaluation's reliance on IID data distribution.

Yin and Zeng (2024) [164] tackle the heightened vulnerability of non-IID FL to Byzantine data poisoning attacks, where existing defenses often fail. Their proposed framework, FLDA, introduces a dual defense mechanism operating on both the client and server. The core client-side strategy is local data augmentation via mixup, which serves to both enhance model robustness against poisoned samples and mitigate the statistical heterogeneity from non-IID data. This is complemented server-side by a gradient detection strategy. The server employs k-means clustering on gradient history to identify and reduce the aggregation weights of malicious clients, thereby minimizing their impact. An incentive mechanism is also included to encourage clients to perform the augmentation. Empirical results on EMNIST, Fashion-MNIST, and CIFAR-10 show FLDA effectively defends against label-flipping and untargeted attacks. The gradient detection component is shown to be highly effective, increasing accuracy by more than 12%. While the framework maintains stability across various non-IID degrees and attack intensities, its defense effectiveness is noted to decline slightly with very small client datasets or high attack severity.

Li et al. (2024) [165] propose EPPRFL to simultaneously defend against Byzantine (poisoning) attacks from clients and privacy leaks from honest-but-curious servers. The framework's novelty lies in its Filtering & Clipping (F&C) detection method, which securely identifies outliers using squared Euclidean distance against a historical median benchmark. To ensure efficiency, it integrates update downsampling (reducing dimensionality) and a dual-server additive secret sharing model, for which the authors customize secure protocols (comparison, median, distance, clipping) to protect all intermediate values. On MNIST, CIFAR-10, FEMNIST, and CelebA datasets, EPPRFL demonstrates robust defense against noise, label-flipping, and sign-flipping attacks. Analytically, its median-based statistical approach proves more stable than baselines like Byzantine-resilient secure federated learning (BREA) [166] and lightweight and secure federated learning (LSFL) [167], maintaining higher accuracy even in high-adversary scenarios with 40% attackers. Client-side efficiency is a key strength, where computation and communication are 50% lower than LSFL. While server-side complexity is optimized to $O(dN)$, this overhead still scales linearly with the number of clients. However, a significant limitation is that EPPRFL provides only partial mitigation against adaptive backdoor attacks like 3DFed.

A central challenge in federated learning is simultaneously defending against poisoning attacks and privacy-inference attacks. Privacy-preserving techniques, such as encryption, often hide the gradients, which makes robust aggregation methods that need to inspect gradients impossible. The reviewed works propose novel architectures and cryptographic protocols to solve this efficiency and security dilemma. Both Shen et al. [163] and Li et al. (EPPRFL) [165] introduce multi-server architectures (three-server and dual-server, respectively) to offload the intense computational work from resource-limited IoT devices. These

approaches differ in their cryptographic foundations. Li et al. (PFLS) build their system on multidimensional homomorphic encryption (HE) [162]. In contrast, Shen et al. [163] and Li et al. (EPPRFL) [165] both opt for Additive Secret Sharing (ASS), arguing it is far more efficient. Shen et al. report their ASS-based method is 141 times faster than HE approaches [163], and Li et al. (EPPRFL) note their client-side overhead is 50% lower than other baselines [165].

Beyond the cryptographic layer, these papers implement different server-side defense logic. Li et al. (PFLS) [162] and Li et al. (EPPRFL) [165] both use Euclidean distance to detect outliers, but EPPRFL compares against a historical median benchmark rather than a masked base gradient. Shen et al. also use a median benchmark but calculate robustness using the Pearson correlation coefficient [163]. Yin and Zeng propose a completely different approach. They use a client-side mixup (data augmentation) strategy to make models inherently robust. This is paired with a server-side k-means clustering method on gradient history to down-weight malicious clients [164]. This focus on data augmentation also directly addresses the critical challenge of non-IID data, a problem that can cause benign clients to appear malicious. While the other schemes also test against non-IID data, the defense from Yin and Zeng is the only one explicitly designed to mitigate it [164].

Table 12 summarizes various backdoor and poisoning-attack defenses in federated learning, detailing each defense's capability, architecture, mitigation effectiveness, accuracy, cost, and key trade-offs.

Table 12: Summary of defenses against backdoor and poisoning attacks in federated learning

Defense	Capability	Architecture /Access	Mitigation (ASR)	Accuracy	Cost	Trade-offs/Notes
FLSAD [158]	Detect & repair backdoors	Model-layer self-attention	ASR ≤ 2%	MNIST > 93%, CIFAR-10 > 86%	Medium-High	No clean data required; cost increases with trigger size
RFed [161]	Robust aggregation	Dual-server, attention	ASR ≤ 4%	Maintained	Linear $O(nd)$, extra comm.	Secure for ≤50% malicious clients; needs root benchmark
PFLS [162]	Outlier detection	Single-server, HE-based	100% detection	ACC ± 0.5%–15%	Linear compute; extra comm.	Privacy-preserving; communication tuning needed
SPEFL [163]	Robust aggregation	Three-server, additive secret sharing	ASR ≤ 50% malicious	Maintained	$O(n)$, fast (141× HE)	Efficient for IoT; requires multi-server; limited non-IID eval
FLDA [164]	Client mixup + gradient detection	Single-server	Label-flip/untargeted mitigated	ACC +12%	Low-Moderate	Handles non-IID; drops with very small clients

(Continued)

Table 12 (continued)

Defense	Capability	Architecture /Access	Mitigation (ASR)	Accuracy	Cost	Trade-offs/Notes
EPPRFL [165]	Filtering & Clipping (F&C)	Dual-server, additive secret sharing	Robust to 40% attackers	Maintained	Linear $O(dN)$; 50% lower client overhead	Partial mitigation for adaptive backdoors; privacy preserved

8.2.4 Audit and Mask-Based Poisoning Detection Methods

Basak and Chatterjee (2024) [168] propose DPAD (Data Poisoning Attack Defense) to protect Federated Learning (FL) systems from data poisoning attacks originating from Byzantine (malicious) clients, under an honest server assumption. The central problem is the global model's vulnerability to performance degradation when malicious clients submit poisoned updates. DPAD introduces an audit-based verification process where the server uses a public dataset to construct a temporary audit model, calculates a poison rate for each client update, and compares it against a KL divergence-based tolerance threshold. Experiments on the Heart Disease Dataset (289,865 samples after preprocessing) demonstrate DPAD successfully defends against attacks with 10-40% compromised users, achieving 72%–73% accuracy, 0.27–0.28 MSE, and 0.71–0.73 AUC. DPAD consistently ranks first (tying with APFed and, at 10% attackers, ADFL) and outperforms LoMar and ADFL at higher attack percentages. However, auditing every client update imposes computational overhead on the server. Furthermore, the paper notes that if the server becomes semi-honest or curious (violating the initial assumption), the framework's security could be compromised.

Feng et al. (2024) [169] introduce DPFLA to defend Federated Learning (FL) against data poisoning attacks from Byzantine (malicious) participants under a semi-honest (honest-but-curious) server assumption. The central challenge is that defenses either expose raw gradients, risking privacy, or employ heavy encryption/differential privacy that degrades accuracy. DPFLA uses a Trusted Authority to distribute random seeds, enabling participants and servers to generate orthogonal masking matrices locally. Participants apply removable masks to final-layer neuron gradients before uploading. The server aggregates masked gradients, applies SVD for lossless dimensionality reduction, then uses K-means clustering (K = 2) with Silhouette Coefficient validation to distinguish malicious from benign updates. Experiments on MNIST and CIFAR10 datasets demonstrate DPFLA detects poisoning attacks even with 40% malicious participants, achieving >95% detection accuracy while maintaining high test accuracy (e.g., 95.2%–98.7% on MNIST and 71.2%–81.8% on CIFAR10) and outperforming FedAvg, Median, TMean, FoolsGold, and MKrum. However, clustering complex data presents challenges, and the method focuses primarily on label-flipping and backdoor attacks.

Liu et al. (2025) [170] introduce DefendFL, a privacy-preserving federated learning scheme countering poisoning attacks from honest-but-curious servers and Byzantine (malicious) adversaries. Employing collinearity masks, DefendFL preserves gradient direction while obscuring magnitude, enabling secure aggregation, which in mask-based methods does not result in model accuracy loss. Users split gradients into two vectors, masked separately, with public commitments ensuring integrity. A dual-server architecture (AS and MS) performs cosine similarity detection, where suspicious gradients are excluded or weighted down in non-IID settings. Evaluated on MNIST and CIFAR-10 datasets with 20 users, DefendFL achieves 10%–20%

higher accuracy than weight-reduction (sub-models) and noise-addition methods when facing 25%–50% malicious users at 30%–50% poisoning rates. It also demonstrates over 60% faster detection than HE and secret-sharing schemes. Strengths include precise detection, robust privacy, and computational efficiency (operations limited to addition, multiplication, and hash). Limitations involve the non-colluding server requirement, a minimum of three valid users per round, and challenges in distinguishing malicious gradients in non-IID data.

This set of studies examines server-side defenses designed to detect malicious updates prior to aggregation, yet they differ substantially in both detection philosophy and privacy implications. The principal methodological distinction is that DPAD [168] applies a functional audit, evaluating the effect of each update on a public dataset, whereas DPFLA [169] and DefendFL [170] adopt statistical property analysis, focusing on the inherent characteristics of the updates themselves. These statistical methods differ further: DPFLA clusters updates in a latent space derived from masked gradients, while DefendFL measures the directional alignment (cosine similarity) of an update relative to its peers. The most critical trend observed is the tight integration of privacy-preserving techniques into the detection logic itself. DPAD represents a more traditional approach, requiring a trusted server to inspect plaintext updates for its audit, whereas DPFLA and DefendFL are private-by-design. They perform their detection logic, clustering, and similarity checks, respectively, directly on masked gradients, ensuring the server never needs to access the raw updates. This represents a significant maturation of FL defense design, moving from security-at-the-cost-of-privacy to solutions that achieve both simultaneously, albeit with the added architectural complexity of trusted authorities or dual-server models.

8.3 Coordinated Jamming and Poisoning Attacks in Wireless FL

Federated Learning (FL) enables collaborative model training across distributed clients while preserving privacy by exchanging model parameters rather than raw data [171]. When FL is deployed over wireless infrastructures, it benefits from ubiquitous connectivity but becomes tightly coupled to the characteristics of the communication medium [172,173]. This coupling enlarges the attack surface: adversaries can mount coordinated campaigns that span both the physical (communication) and application (learning) layers. Unlike isolated jamming [174] or poisoning [175] attacks, coordinated strategies exploit cross-layer dependencies to amplify their effect and may cause severe degradation of the global model or even prevent convergence [176]. For example, selective jamming can censor or delay high-quality honest updates so that poisoned contributions exert disproportionate influence during aggregation [176], while poisoning attacks can be crafted to leverage or provoke communication impairments that further destabilize learning [173].

Most existing defenses treat jamming and poisoning separately and are therefore likely to be inadequate when adversaries coordinate across layers. Effective protection against this class of threats requires a holistic understanding of the joint dynamics of wireless channels and distributed learning, together with integrated detection and mitigation mechanisms that reason across both domains [176]. This section synthesizes the unique challenges posed by coordinated attacks, surveys detection and mitigation strategies at the communication and learning layers, advocates cross-layer integration, identifies open research gaps, and outlines promising directions for securing wireless FL systems. The primary technical difficulty is reliably distinguishing coordinated malicious behavior from benign phenomena such as channel stochasticity and legitimate data heterogeneity.

8.3.1 Unique Challenges for Detection and Mitigation

Coordinated jamming-poisoning campaigns are difficult to detect and mitigate because radio dynamics and learning dynamics interact in ways that obscure malicious intent. First, the wireless medium is

intrinsically stochastic and non-stationary: multipath fading, mobility, and transient congestion produce fluctuations in RSSI, SINR, and packet delivery ratio (PDR) that closely resemble effects induced by interference, complicating signal-level discrimination and increasing false positives or negatives when decisions rely on PHY metrics alone [174,177,178]. Second, FL typically operates over non-IID client data: benign gradient heterogeneity can cause honest updates to diverge substantially from the population, and carefully crafted, low-magnitude poisoning updates can be concealed within this natural variance, evading simple outlier detectors [171,175,176]. Third, the synchronized, round-based operation of FL creates predictable timing windows that attackers may exploit: passive monitoring or side channels can reveal aggregation schedules or high-value transmission opportunities, allowing adversaries to time jamming bursts so as to selectively censor or delay particular honest contributors while conserving energy and reducing detectability [173,176]. Fourth, communication unreliability and decentralized trust mechanisms provide additional vectors: packet loss and retransmissions, amplified by jamming, distort participation statistics and bias aggregation, and in decentralized FL variants (for example, blockchain-coordinated FL), jamming can disrupt consensus or block propagation, undermining provenance and global model integrity [173,176]. These interdependent difficulties show that layer-specific defenses are generally insufficient. Robust protection, therefore, requires cross-layer reasoning that jointly models communication uncertainty and learning dynamics under adversarial constraints.

8.3.2 Detection Mechanisms

Robust detection of coordinated attacks depends on fusing complementary evidence from the physical and learning planes. At the physical plane, lightweight anomaly detectors monitor trends in RSSI, SINR, PDR, and retransmission counts to provide early indicators of abnormal interference. More discriminative analyses use spectral correlation functions or I/Q spectrogram representations to differentiate jammer classes (constant, reactive, deceptive) and support classifier models such as convolutional neural networks or variational autoencoders trained on RF features [174,177,178]. Federated training of RF classifiers preserves privacy and improves spatial generalization across heterogeneous deployments [177]. Such RF-centric approaches, however, require representative PHY traces, incur edge compute costs, and remain vulnerable to adaptive adversaries that mimic benign signal characteristics.

At the learning plane, anomaly detection examines submitted model updates for statistical irregularities. Similarity metrics (cosine similarity, Euclidean distance), clustering of update vectors, and robust statistics (coordinate-wise median, trimmed mean) flag deviations from expected patterns; impact-based tests that measure an update's effect on a held-out validation set can reveal contributions that consistently harm global performance [171,176,179,180]. These techniques work well for obvious or large-magnitude manipulations but struggle with non-IID heterogeneity and colluders who distribute subtle perturbations across clients.

Cross-layer correlation between PHY and learning indicators is conceptually attractive and, we argue, promising for improving detection fidelity. For instance, synchronized SINR degradations among a geographically localized set of clients that coincide with an unusual pattern of submitted model updates could strengthen evidence for coordinated interference and poisoning. Likewise, the temporal alignment of radio anomalies and suspicious update arrivals is a plausible indicator of coordination. Nevertheless, the specific correlation patterns, decision thresholds, and empirical validation of these cross-layer indicators remain largely hypothesized in the current FL literature [174,176]; direct experimental validation is limited. We therefore present cross-layer correlation here as a hypothesis-driven detection strategy motivated by the challenges rather than as an established, universally validated technique.

Motivated by these considerations, a pragmatic, hypothesis-driven detection architecture is staged: continuous, low-cost PHY monitoring at edge nodes; periodic reporting of compact channel and participation

summaries to the server; and server-side fusion that triggers more compute-intensive, high-fidelity analyses (for example, spectrogram inspection or validation-impact tests) only when cross-layer signals indicate elevated risk. Elements of this multi-tiered approach appear across the literature (e.g., edge monitoring and federated RF classifiers [177], and server-side aggregation logic [176]); the exact staged pipeline we describe is a reasoned proposal designed to balance timeliness, resource constraints, and detection fidelity and warrants targeted experimental evaluation.

8.3.3 Mitigation Mechanisms

Mitigation must both limit immediate damage from active interference and preserve long-term model integrity; defenses therefore operate at the communication plane, the aggregation plane, and via hybrid cross-layer policies.

At the communication level, proactive techniques such as channel diversity (frequency hopping), spread-spectrum, and directional transmission (beamforming) reduce susceptibility to narrowband or spatially localized jammers. Reactive tactics, opportunistic relaying, route diversification, and temporary prioritization of clients on more robust channels are plausible operational counters that may be applied when interference is detected [174,181]. Although related path-selection and route-adaptation concepts are examined in the literature [174], explicit experimental demonstrations of relaying and per-client prioritization as anti-jamming measures in FL are partial; accordingly, we present these tactics as plausible mitigations that merit targeted validation in FL settings.

Game-theoretic formulations and reinforcement-learning (RL) methods provide adaptive frameworks for channel and power allocation under uncertainty [182,183], and federated deep RL enables distributed learning of mitigation policies without centralizing raw PHY observations [184]. These approaches are promising for decentralized deployment but impose trade-offs in training overhead, convergence behavior, and safe exploration that must be managed.

At the aggregation layer, robust statistical rules (coordinate-wise median, trimmed mean, geometric median) and selection strategies (Krum, Multi-Krum) reduce the influence of outlying updates; contribution-aware schemes (validation-based weighting, reputation systems, approximate Shapley values) dynamically scale client influence according to measured utility [171,176,179]. Gradient clipping and normalization are standard adversarial-ML primitives for constraining perturbation magnitude; while their direct evaluation as primary aggregation defenses in the specific FL studies cited here is limited, they remain practical, low-cost primitives that can be combined with robust aggregation to reduce attack impact and should be supported by broader adversarial-ML references. Cryptographic techniques (homomorphic encryption, secure multiparty computation) and verifiable ledgers (blockchain) provide confidentiality and auditability but impose substantial computational and communication overheads and remain sensitive to availability attacks that disrupt consensus [173,175,178]. Hybrid mitigation policies, such as temporarily applying conservative aggregation rules while prioritizing protected channels for high-value clients during detected jamming, present logical, jointly responsive strategies.

9 Challenges and Future Directions

Federated Learning (FL) has evolved from a promising concept to a complex field of privacy-preserving machine learning. The literature shows steady progress across VFL, HFL, FTL, and PFL. It also shows that many practical problems remain. The next phase of research must move beyond isolated proofs of concept. It must produce systems that are secure, robust, efficient, and auditable in the messy conditions of real deployments. Below, we outline the challenges and concrete research directions.

9.1 Heterogeneity at Multiple Levels

Heterogeneity at multiple levels remains unsolved. VFL advances such as MMVFL, PraVFed, and HeteroVFL address different slices of the problem. Feng et al. [84] extend VFL to multi-party multi-class tasks. Wang et al., [85] let clients run different local architectures. Zhang et al. [86] use prototype aggregation to handle distributional mismatch. These are important but partial answers. Real systems must cope with client heterogeneity in data, model architecture, compute budget, network quality, privacy requirements, and label overlap simultaneously. Designing unified frameworks that adapt to all these axes without exploding communication, computation, or tuning effort is a core research priority. Promising approaches include modular architectures that separate representation learning, task heads, and aggregation rules, and meta-learning schemes that learn how to assign modules to clients based on measured heterogeneity.

9.2 Privacy Protection

Privacy protection is no longer only about adding differential noise. The field has shifted from raw gradient sharing to higher-level abstractions such as embeddings, prototypes, and pseudo-labels. That shift reduces some attack vectors while creating new ones. PraVFed [85] and HeteroVFL [86] explicitly add differential privacy for protection. Homomorphic encryption based protocols and substitution techniques offer alternative guarantees [88,89]. Each approach has limits. Differential privacy can degrade accuracy at realistic privacy budgets, homomorphic encryption is slow for deep non-linear models, and simple substitution only defeats narrow attacks. What the field lacks is a principled measurement of how much private information flows through various abstractions. We need empirical benchmarks and theoretical bounds that quantify leakage from embeddings, prototypes, and pseudo-labels. We also need adaptive privacy controllers that allocate protection where risk is highest. These controllers should consider the task sensitivity, adversary model, available compute, and acceptable utility loss. Hybrid designs that combine light cryptography for high risk components and lightweight perturbation elsewhere may hit better points on the privacy-utility-efficiency frontier.

9.3 Communication and Systems Scalability

Communication and systems scalability are still a bottleneck. Compression, client selection, and feature selection reduce cost in isolation. EF-VFL [87] shows faster convergence under error feedback compression. Fed-FiS and FedCSR style methods reduce per-round cost by pruning features or selecting clients [99,100]. But none of these techniques has been validated end-to-end at internet scale while preserving privacy and robustness. Practical deployments must coordinate compression, selective participation, and adaptive syncing across unstable networks and millions of devices. Research should move from single-technique papers toward cross-layer co-design. That means joint optimization of model architecture, compression schedule, synchronization policy, and privacy budget. It also means developing adaptive protocols that measure network and compute conditions and change behavior at runtime. Hierarchical aggregation using edge servers and semi-synchronous scheduling are promising system-level patterns [134,137]. Evaluations should include communication cost, wall clock time, energy, and accuracy under realistic churn.

9.4 Robustness against Adaptive Adversaries

Robustness against adaptive adversaries remains underdeveloped. Attacks have evolved from early gradient inversion to generation-based and meta-learning attacks that generalize across defenses [148,149,155]. Poisoning strategies now include Sybil amplification and property-inference-assisted poisoning [13,153,154]. Defenses span from robust aggregation to post-hoc repair [158,161]. The observation shows two gaps. First, most defenses are tailored to narrow attack models and break under slightly different assumptions.

Second, evaluations often test one attack at a time and use synthetic poisoners. The field needs standardized threat models and benchmark suites that expose defenses to adaptive attackers. It also needs defenses with formal guarantees when possible. Certified robustness for certain classes of poisoning or leakage attacks is an ambitious but valuable direction. Combining lightweight runtime anomaly detectors, history-based baselines, and periodic audit mechanisms may produce practical layered defenses that balance cost and security [162,163,165].

9.5 Coordinated Cross-Layer Attacks in Wireless FL

The inclusion of coordinated jamming and poisoning attacks in wireless FL reveals a critical vulnerability class that traditional defenses fail to address. The literature demonstrates that adversaries can exploit the tight coupling between communication and learning layers to amplify attack effectiveness [173,176]. The tight coupling of the communication medium and the learning process creates an enlarged attack surface where adversaries can, for example, selectively jam honest clients to increase the influence of poisoned updates [173,176]. The core research challenge is that distinguishing this coordinated malicious behavior from benign phenomena, such as radio channel stochasticity [174,177] and legitimate non-IID data heterogeneity [171,175], is exceptionally difficult. Future research must, therefore, focus on holistic, cross-layer defenses [176]. This requires detection mechanisms that fuse evidence from the physical plane (e.g., SINR, PDR) and the learning plane (e.g., gradient anomalies) [177,181]. Mitigation must also be integrated, combining communication-level tactics, such as federated deep RL for resource allocation [184], with learning-level robust aggregation [176,179].

9.6 Personalization and Fairness

Personalization and fairness require new principles and metrics. PFL methods such as pFedSD [115], pFedGate [116], pFedKT [117], and pFedSV [118] show a range of personalization strategies. FairDPFL-SCS [119] and CA-PFL [121] begin to address fairness across clients [115–119]. Yet fairness metrics from centralized ML do not translate cleanly to federated systems. In FL, the data distribution and participation rates are heterogeneous. A model that is fair on average can hurt small or low-participation groups. Future work must produce fairness definitions that capture both per-client utility and group equity under sparse participation. Aggregation rules and client selection must be fairness-aware. Incentive schemes should avoid rewarding only high-capacity participants. Mechanism design and game theory can inform incentive-compatible aggregation and payment methods that reflect contribution while preserving fairness.

9.7 Evaluation, Datasets, and Reproducibility

Evaluation, datasets, and reproducibility are weak links. Many papers report results on bespoke splits or small benchmarks that do not capture realistic non-IID conditions. Domain-specific systems for XGBoost, medical imaging, smart grids, and IIoT show real value but also a proliferation of incompatible evaluations [90,91,93,113]. The community needs federated benchmarks that simulate realistic heterogeneity in scale, overlap, and network conditions. Benchmarks must measure privacy leakage, communication cost, wall clock time, energy, fairness, and robustness. Open-source reference implementations and reproducible scripts are needed to make comparisons meaningful.

9.8 Theory for Federated Settings

Theory for federated settings must catch up. Convergence analyses often assume IID data, synchronous updates, or full gradients. Real designs use compression, partial layer updates, representation sharing, asynchronous updates, and privacy noise. Theory must provide convergence and generalization bounds for

realistic protocols. That includes bounds for hybrid VFL-HFL systems, bounds under client sampling, and rates when using learned compression. Theory should also formalize leakage from representation sharing. Information-theoretic analyses can give lower bounds on what embeddings reveal about raw data. Such bounds will guide mechanism design and privacy accounting.

9.9 Domain Adaptation and Transfer

Domain adaptation and transfer remain practical levers. FTL approaches and distributed GAN-based augmentation have shown ways to tackle insufficient overlap [91,108]. Transfer learning accelerates convergence in resource-constrained devices [106,107]. The research challenge is systematization. Which transfer approach suits which domain? How to combine transfer and personalization without opening new leakage vectors [110]? How to handle continual distribution shift in long lived systems such as smart grids and healthcare devices? Solutions will need adaptive transfer modules, robust domain clustering, and continual learning techniques that resist catastrophic forgetting while preserving privacy.

9.10 Secure Computation Choices

Secure computation choices must be pragmatic. Homomorphic encryption and MPC give strong guarantees but at a cost [89]. Compressed sensing, learned autoencoders, and secret sharing offer lighter-weight options [122,151,152]. Hardware-based trust, such as Trusted Execution Environment (TEE), is attractive but raises deployment and trust questions. Future work should compare these approaches under real workloads and define hybrid stacks that use expensive cryptography only for small, high-risk components while using efficient perturbation elsewhere.

9.11 Operational Engineering, Regulatory, and Governance Issues

Operational engineering matters as much as algorithms. Production-grade FL needs monitoring, debugging, logging, and rollback mechanisms that preserve privacy. Debugging a federated model is hard when raw data and intermediate representations are private. Tooling that supports privacy-preserving testing, federated unit tests, and interpretable audit trails is needed. Interpretability methods adapted to federated settings will help domain experts validate models, especially in regulated fields such as healthcare and finance. Regulatory, legal, and governance issues will shape research directions. Data sovereignty, cross-border rules, and consent constraints vary by jurisdiction. Research must build compliance into protocols. Auditability and tamper evidence are important features. Blockchain and distributed ledger ideas offer one route, but they add cost and complexity [104,111]. Work that aligns technical properties to legal definitions of processing, consent, and accountability will help adoption.

9.12 Sustainability and Cost

Sustainability and cost must be part of the research agenda. Edge and mobile training waste energy. Methods that cut rounds, reduce device on time, or shift computing to more efficient nodes will lower the carbon footprint. Energy-aware model design and scheduling will matter as FL moves from prototypes to continuous production.

9.13 Concrete Research Directions

Concrete research directions follow from these challenges. Developing modular FL kernels that let practitioners plug in privacy, compression, and personalization modules. Building adaptive privacy controllers that use measured leakage risk to tune noise and cryptographic fallback. Investing in benchmark suites that cover cross silo and cross-device scenarios. Analyzing certified defenses and realistic attacker

models that combine collusion, Sybil nodes, and adaptive learning. Formulating fairness definitions that reflect client participation and group equity. Producing theoretical bounds for convergence under the full stack of compression, asynchrony, and privacy noise. Designing hybrid secure stacks that mix HE, MPC, secret sharing, and lightweight perturbation in a cost-aware manner. Building open reference implementations with monitoring, debugging, and audit support while coordinating with legal scholars to make FL designs that satisfy common regulatory requirements.

10 Conclusion

This review synthesizes advances across VFL, HFL, FTL, and PFL, showing the field has moved from proofs of concept to practical, domain-aware systems. Progress is nonetheless fragmented, and many solutions are narrow, leaving unmet needs in multi-axis heterogeneity, measurable leakage from representation sharing, and scalable, secure protocols for real deployments. Priority directions are clear. We need modular, adaptive FL kernels, standard benchmarks and threat models, adaptive privacy controllers that tune protection by risk, high-stakes domains, theoretical guarantees for realistic protocols, and operational tooling for auditing and compliance. Delivering trustworthy federated systems for healthcare, finance, energy, and similar high-stakes domains will require interdisciplinary work across machine learning systems, cryptography, law, and economics.

Acknowledgement: None.

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

Author Contributions: The authors confirm their contributions to this manuscript as follows: conceptualization and study design: Faisal Mahmud, Fahim Mahmud, Rashedur M. Rahman; literature review and methodology development: Faisal Mahmud, Fahim Mahmud; data curation and analysis: Faisal Mahmud, Fahim Mahmud; writing—original draft preparation: Faisal Mahmud, Fahim Mahmud; writing—review and editing: Rashedur M. Rahman; supervision and project administration: Rashedur M. Rahman. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All reviewed materials are publicly available and cited, primarily sourced from major academic repositories and publishers including IEEE Xplore, ACM Digital Library, ScienceDirect (Elsevier), SpringerLink, AAAI, Nature, and the arXiv repository.

Ethics Approval: Not applicable.

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

References

1. Zhao X, Wang L, Zhang Y, Han X, Deveci M, Parmar M. A review of convolutional neural networks in computer vision. Artif Intell Rev. 2024;57(4):57–99. doi:10.1007/s10462-024-10721-6.
2. Xiao F, Cai S, Chen G, Jagadish HV, Ooi BC, Zhang M. VecAug: unveiling camouflaged frauds with cohort augmentation for enhanced detection. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; New York, NY, USA: Association for Computing Machinery; 2024. p. 6025–36. doi:10.1145/3637528.3671527.
3. Khogali HO, Mekid S. The blended future of automation and AI: examining some long-term societal and ethical impact features. Technol Soc. 2023;73(2):102232. doi:10.1016/j.techsoc.2023.102232.
4. Quach S, Thaichon P, Martin KD, Weaven S, Palmatier RW. Digital technologies: tensions in privacy and data. J Acad Mark Sci. 2022;50(6):1299–323. doi:10.1007/s11747-022-00845-y.

5. Wu C. Data privacy: from transparency to fairness. Technol Soc. 2024;76(1):102457. doi:10.1016/j.techsoc.2024.102457.
6. Narayanan A, Shmatikov V. Robust de-anonymization of large sparse datasets. In: 2008 IEEE Symposium on Security and Privacy (SP 2008); 2008 May 18–22; Oakland, CA, USA. Piscataway, NJ, USA: IEEE. p. 111–25. doi:10.1109/SP.2008.33.
7. Colangelo G, Maggiolino M. Data accumulation and the privacy-antitrust interface: insights from the Facebook case. International Data Privacy Law. 2018;8(3):224–39. doi:10.1093/idpl/ipy018.
8. Voigt P, von dem Bussche A. The EU general data protection regulation (GDPR): a practical guide. Cham, Switzerland: Springer; 2017. doi:10.1007/978-3-319-57959-7.
9. Zhang F, Shuai Z, Kuang K, Wu F, Zhuang Y, Xiao J. Unified fair federated learning for digital healthcare. Patterns. 2024;5(1):100907. doi:10.1016/j.patter.2023.100907.
10. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Singh A, Zhu J, editors. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Vol. 54; London, UK: PMLR; 2017. p. 1273–82.
11. Wang Q, Dong H, Huang Y, Liu Z, Gou Y. Blockchain-enabled federated learning for privacy-preserving non-IID data sharing in industrial internet. Comput Mater Contin. 2024;80(2):1967–83. doi:10.32604/cmc.2024.052775.
12. Konečný J, McMahan HB, Yu FX, Suresh AT, Bacon D, Richtárik P. Federated learning: strategies for improving communication efficiency. arXiv:1610.05492. 2018.
13. Wang Z, Huang Y, Song M, Wu L, Xue F, Ren K. Poisoning-assisted property inference attack against federated learning. IEEE Trans Dependable Secure Comput. 2023;20(4):3328–40. doi:10.1109/TDSC.2022.3196646.
14. Boenisch F, Dziedzic A, Schuster R, Shamsabadi A, Shumailov I, Papernot N. When the curious abandon honesty: Federated learning is not private. In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P); 2023 Jul 3–7; Delft, Netherlands. Piscataway, NJ, USA: IEEE. p. 175–99. doi:10.1109/EuroSP57164.2023.00020.
15. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends Mach Learn. 2021;14(1–2):1–210.
16. Yurdem B, Kuzlu M, Gullu MK, Catak FO, Tabassum M. Federated learning: overview, strategies, applications, tools and future directions. Heliyon. 2024;10(19):e38137. doi:10.1016/j.heliyon.2024.e38137.
17. Vajrobol V, Saxena G, Pundir A, Singh S, Gaurav A, Bansal S, et al. A comprehensive survey on federated learning applications in computational mental healthcare. Comput Model Eng Sci. 2025;142(1):49–90. doi:10.32604/cmes.2024.056500.
18. Wang J, Liu Z, Yang X, Li M, Lyu Z. The internet of things under federated learning: a review of the latest advances and applications. Comput Mater Contin. 2025;82(1):1–39. doi:10.32604/cmc.2024.058926.
19. Khanh QV, Chehri A, Dang VA, Minh QN. Federated learning approach for collaborative and secure smart healthcare applications. IEEE Trans Emerg Top Comput. 2024;13(1):68–79. doi:10.1109/TETC.2024.3473911.
20. Pei J, Liu W, Li J, Wang L, Liu C. A review of federated learning methods in heterogeneous scenarios. IEEE Trans Consum Electron. 2024;70(3):5983–99. doi:10.1109/TCE.2024.3385440.
21. Nezhadsistani N, Moayedian NS, Stiller B. Blockchain-enabled federated learning in healthcare: survey and state-of-the-art. IEEE Access. 2025;13(1999):119922–45. doi:10.1109/ACCESS.2025.3587345.
22. Cai Z, Chen J, Fan Y, Zheng Z, Li K. Blockchain-empowered federated learning: benefits, challenges, and solutions. IEEE Trans Big Data. 2025;11(5):2244–63. doi:10.1109/TBDATA.2025.3541560.
23. Tariq A, Serhani MA, Sallabi FM, Barka ES, Qayyum T, Khater HM, et al. Trustworthy federated learning: a comprehensive review, architecture, key challenges, and future research prospects. IEEE Open J Commun Soc. 2024;5:4920–98. doi:10.1109/OJCOMS.2024.3438264.
24. Kang Y, Gu H, Tang X, He Y, Zhang Y, He J, et al. Optimizing privacy, utility, and efficiency in a constrained multi-objective federated learning framework. ACM Trans Intell Syst Technol. 2024;15(6):1–33. doi:10.1145/3701039.
25. Rao B, Zhang J, Wu D, Zhu C, Sun X, Chen B. Privacy inference attack and defense in centralized and federated learning: a comprehensive survey. IEEE Trans Artif Intell. 2025;6(2):333–53. doi:10.1109/TAI.2024.3363670.
26. Upreti D, Yang E, Kim H, Seo C. A comprehensive survey on federated learning in the healthcare area: concept and applications. Comput Model Eng Sci. 2024;140(3):2239–74. doi:10.32604/cmes.2024.048932.

27. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020;10(1):1–12. doi:10.1038/s41598-020-69250-1.
28. Lai P, Zhang M, Tang Y, Yue Y, Di F. VPAFL: verifiable privacy-preserving aggregation for federated learning based on single server. Comput Mater Contin. 2025;84(2):2935–57. doi:10.32604/cmc.2025.065887.
29. Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019—2019 IEEE International Conference on Communications (ICC); Piscataway, NJ, USA: ICC; 2018. p. 1–7. doi:10.1109/ICC.2019.8761315.
30. Chen D, Gao D, Xie Y, Pan X, Li Z, Li Y, et al. FS-REAL: towards real-world cross-device federated learning. In: KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; New York, NY, USA: ACM; 2023. p. 3829–41. doi:10.1145/3580305.3599829.
31. Huba D, Nguyen J, Malik K, Zhu R, Rabbat MG, Yousefpour A, et al. Papaya: practical, private, and scalable federated learning. In: Proceedings of Machine Learning and Systems [Internet]. Vol. 4. MLSys Conference Committee; 2022. p. 814–32. [cited 2025 Nov 2]. Available from: https://proceedings.mlsys.org/paper_files/paper/2022/hash/a8bc4cb14a20f20d1f96188bd61eec87-Abstract.html.
32. Li Q, Diao Y, Chen Q, He B. Federated learning on non-IID data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE); Piscataway, NJ, USA: IEEE; 2021. p. 965–78. doi:10.1109/icde53745.2022.00077.
33. Reddi SJ, Charles Z, Zaheer M, Garrett Z, Rush K, Konečný J, et al. Adaptive federated optimization. In: International Conference on Learning Representations. arXiv:2003.00295. 2021. doi:10.48550/arXiv.2003.00295.
34. Ye H, Liang L, Li GY. Decentralized federated learning with unreliable communications. IEEE J Sel Top Signal Process. 2022;16(3):487–500. doi:10.1109/jstsp.2022.3152445.
35. Stripelis D, Ambite JL. Federated learning over harmonized data silos. In: Shaban-Nejad A, Michalowski M, Bianco S, editors. Artificial intelligence for personalized medicine: promoting healthy living and longevity. Cham, Switzerland: Springer Nature; 2023. p. 27–41. doi:10.1007/978-3-031-36938-4_3.
36. Zeng Y, Teng S, Xiang T, Zhang J, Mu Y, Ren Y, et al. A client selection method based on loss function optimization for federated learning. Comput Model Eng Sci. 2023;137(1):1047–64. doi:10.32604/cmes.2023.027226.
37. Sun Y, Kountouris M, Zhang J. How to collaborate: towards maximizing the generalization performance in cross-silo federated learning. IEEE Trans Mob Comput. 2025;24(5):3211–22. doi:10.1109/TMC.2024.3509852.
38. Yashwanth M, Nayak GK, Singh A, Simmhan Y, Chakraborty A. Adaptive self-distillation for minimizing client drift in heterogeneous federated learning. Trans Mach Learn Res. 2024 [cited 2025 Oct 20]. Available from: https://openreview.net/forum?id=K58n87DE4s.
39. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. SCAFFOLD: stochastic controlled averaging for federated learning. In: Daumé III H, Singh A, editors. Proceedings of the 37th International Conference on Machine Learning. Vol. 119; London, UK: PMLR; 2020. p. 5132–43.
40. Sánchez Sánchez PM, Huertas Celdrán A, Martínez Pérez ET, Demeter D, Bovet G, Martínez Pérez G, et al. Analyzing the robustness of decentralized horizontal and vertical federated learning architectures in a non-IID scenario. Appl Intell. 2024;54(8):6637–53. doi:10.1007/s10489-024-05510-1.
41. Liu S, Yu G, Chen X, Bennis M. Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data. IEEE Trans Wirel Commun. 2022;21(10):7852–66. doi:10.1109/TWC.2022.3162595.
42. Ye M, Fang X, Du B, Yuen PC, Tao D. Heterogeneous federated learning: state-of-the-art and research challenges. ACM Comput Surv. 2023;56(3):1–44. doi:10.1145/3625558.
43. Zhang X, Mavromatis A, Vafeas A, Nejabati R, Simeonidou D. Federated feature selection for horizontal federated learning in IoT networks. IEEE Internet Things J. 2023;10(11):10095–112. doi:10.1109/JIOT.2023.3237032.
44. Oh J, Kim S, Yun S-Y. FedBABU: towards enhanced representation for federated image classification. arXiv:2106.06042. 2021.
45. Wang X, Zhang H, Yang M, Wu X, Cheng P. Privacy-preserving collaborative learning: a scheme providing heterogeneous protection. IEEE Internet Things J. 2024;11(2):1840–53. doi:10.1109/JIOT.2023.3289546.

46. Tramèr F, Shokri R, San Joaquin A, Le HM, Jagielski M, Hong S, et al. Truth serum: Poisoning machine learning models to reveal their secrets. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security; New York, NY, USA: ACM; 2022. doi:10.1145/3548606.3560554.
47. Zhang Y, Wu Y, Li T, Zhou H, Chen Y. Vertical federated learning based on consortium blockchain for data sharing in mobile edge computing. Comput Model Eng Sci. 2023;137(1):345–61. doi:10.32604/cmes.2023.026920.
48. Resende A, Aranha DF. Faster unbalanced private set intersection. In: Gritzalis S, Weippl E, editors. Cryptology and network security. Vol. 11124. Cham, Switzerland: Springer; 2018. p. 203–21. doi:10.1007/978-3-662-58387-6_11.
49. Wortsman M, Ilharco G, Gadre S, Roelofs R, Gontijo-Lopes R, Morcos AS, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. arXiv: 2203.05482. 2022. doi:10.48550/arxiv.2203.05482.
50. Wu Y, Xing N. Falcon: a privacy-preserving and interpretable vertical federated learning system. Proc VLDB Endow. 2023;16(10):2471–84. doi:10.14778/3603581.3603588.
51. Zhang C, Liu Z, Xu X, Hu F, Dai J, Cai B, et al. SensFL: privacy-preserving vertical federated learning with sensitive regularization. Comput Model Eng Sci. 2025;142(1):385–404. doi:10.32604/cmes.2024.055596.
52. Zheng Y, Xu S, Wang S, Gao Y, Hua Z. Privet: a privacy-preserving vertical federated learning service for gradient boosted decision tables. IEEE Trans Serv Comput. 2023;16(5):3604–20. doi:10.1109/TSC.2023.3279839.
53. Zou T, Liu Y, Kang Y, Liu W, He Y, Yi Z, et al. Defending batch-level label inference and replacement attacks in vertical federated learning. IEEE Trans Big Data. 2024;10(6):1016–27. doi:10.1109/TBDATA.2022.3192121.
54. Ye M, Shen W, Du B, Snezhko E, Kovalev V, Yuen P. Vertical federated learning for effectiveness, security, applicability: a survey. ACM Comput Surv. 2025;57(9):223. doi:10.1145/3720539.
55. Yin Z, Wang H, Chen B, Zhang X, Lin X, Sun H, et al. Federated semi-supervised representation augmentation with cross-institutional knowledge transfer for healthcare collaboration. Knowl Based Syst. 2024;300(21):112208. doi:10.1016/j.knosys.2024.112208.
56. Liu Y, Kang Y, Xing C, Chen T, Yang Q. A secure federated transfer learning framework. IEEE Intell Syst. 2020;35(4):70–82. doi:10.1109/MIS.2020.2988525.
57. Zhu G, Liu X, Tang S, Niu J. Aligning before aggregating: enabling communication efficient cross-domain federated learning via consistent feature extraction. IEEE Trans Mob Comput. 2024;23(5):5880–96. doi:10.1109/TMC.2023.3316645.
58. Irfan M, Malik KM, Muhammad K. Federated fusion learning with attention mechanism for multi-client medical image analysis. Inf Fusion. 2024;108(7):102364. doi:10.1016/j.inffus.2024.102364.
59. Guo W, Zhuang F, Zhang X, Tong Y, Dong J. A comprehensive survey of federated transfer learning: challenges, methods and applications. Front Comput Sci. 2024;18:186356. doi:10.48550/arXiv.2403.01387.
60. Zhang J, Guo S, Guo J, Zeng D, Zhou J, Zomaya A. Towards data-independent knowledge transfer in model-heterogeneous federated learning. IEEE Trans Comput. 2023;72(10):2888–901. doi:10.1109/TC.2023.3272801.
61. Qiao Y, Le HQ, Zhang M, Adhikary A, Zhang C, Hong CS. FedCCL: federated dual-clustered feature contrast under domain heterogeneity. Inf Fusion. 2025;113(3):102645. doi:10.1016/j.inffus.2024.102645.
62. Wang K, Zhou X, Liang W, Yan Z, She J. Federated transfer learning based cross-domain prediction for smart manufacturing. IEEE Trans Ind Inform. 2022;18(6):4088–96. doi:10.1109/TII.2021.3088057.
63. Qi T, Wu F, Wu C, He L, Huang Y, Xie X. Differentially private knowledge transfer for federated learning. Nat Commun. 2023;14(1):3785. doi:10.1038/s41467-023-38794-x.
64. Zhang Z, He N, Li D, Gao H, Gao T, Zhou C. Federated transfer learning for disaster classification in social computing networks. J Saf Sci Resilience. 2022;3(1):15–23. doi:10.1016/j.jnlssr.2021.10.007.
65. Yang L, Huang J, Lin W, Cao J. Personalized federated learning on non-IID data via group-based meta-learning. ACM Trans Knowl Discov Data. 2022;17(4):1–20. doi:10.1145/3558005.
66. Hard AS, Rao K, Mathews R, Beaufays F, Augenstein S, Eichner H, et al. Federated learning for mobile keyboard prediction. arXiv:1811.03604. 2018.
67. Wu R, Xu J, Zhang Y, Zhao C, Xie Y, Wu Z, et al. Video action recognition method based on personalized federated learning and spatiotemporal features. Comput Mater Contin. 2025;83(3):4961–78. doi:10.32604/cmc.2025.061396.

68. Liu Y, Guo S, Zhang J, Hong Z, Zhan Y, Zhou Q. Collaborative neural architecture search for personalized federated learning. IEEE Trans Comput. 2025;74(1):250–62. doi:10.1109/TC.2024.3477945.
69. Sahu AK, Li T, Sanjabi M, Zaheer M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. arXiv:1812.06127. 2018.
70. Li X, Liu S, Zhou Z, Xu Y, Guo B, Yu Z. ClassTer: mobile shift-robust personalized federated learning via class-wise clustering. IEEE Trans Mob Comput. 2025;24(3):2014–28. doi:10.1109/TMC.2024.3487294.
71. Sun Q. Personalized federated meta-learning based on dynamic clustering. In: Proceedings of 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC); Piscataway, NJ, USA: IEEE; 2024. p. 391–4. doi:10.1109/ICAIRC64177.2024.10899993.
72. Dai R, Shen L, He F, Tian X, Tao D. DisPFL: towards communication-efficient personalized federated learning via decentralized sparse training. arXiv:2206.00187. 2022. doi:10.48550/arxiv.2206.00187.
73. Wei K, Li J, Ma C, Ding M, Chen W, Wu J, et al. Personalized federated learning with differential privacy and convergence guarantee. IEEE Trans Inf Forensics Secur. 2023;18:4488–503. doi:10.1109/TIFS.2023.3293417.
74. Wang Z, Zhang Z, Tian Y, Yang Q, Shan H, Wang W, et al. Asynchronous federated learning over wireless communication networks. IEEE Trans Wirel Commun. 2022;21(10):8415–29. doi:10.1109/TWC.2022.3153495.
75. Zhu H, Zhou Y, Qian H, Shi Y, Chen X, Yang Y. Online client selection for asynchronous federated learning with fairness consideration. IEEE Trans Wirel Commun. 2023;22(4):2493–506. doi:10.1109/TWC.2022.3211998.
76. Gu Y, Bai Y, Xu S. CS-MIA: membership inference attack based on prediction confidence series in federated learning. J Inf Secur Appl. 2022;67(3):103201. doi:10.1016/j.jisa.2022.103201.
77. Wang X, Wu L, Guan Z. GradDiff: gradient-based membership inference attacks against federated distillation with differential comparison. Inf Sci. 2024;658(4):120068. doi:10.1016/j.ins.2023.120068.
78. Balle B, Cherubin G, Hayes J. Reconstructing training data with informed adversaries. In: 2022 IEEE Symposium on Security and Privacy (SP); Piscataway, NJ, USA: IEEE; 2022. p. 1138–56. doi:10.1109/sp46214.2022.9833677.
79. Long Y, Ying Z, Yan H, Fang R, Li X, Wang Y, et al. Membership reconstruction attack in deep neural networks. Inf Sci. 2023;634(2):27–41. doi:10.1016/j.ins.2023.03.008.
80. Gupta P, Yadav K, Gupta BB, Alazab M, Gadekallu T. A novel data poisoning attack in federated learning based on inverted loss function. Comput Secur. 2023;130(9):103270. doi:10.1016/j.cose.2023.103270.
81. Yazdinejad A, Dehghantanha A, Karimipour H, Srivastava G, Parizi R. A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Trans Inf Forensics Secur. 2024;19:6693–708. doi:10.1109/TIFS.2024.3420126.
82. Idrissi MJ, Alami H, El Mahdaouy A, El Mekki A, Oualil S, Yartaoui Z, et al. Fed-ANIDS: federated learning for anomaly-based network intrusion detection systems. Expert Syst Appl. 2023;234(1):121000. doi:10.1016/j.eswa.2023.121000.
83. Song S, Xu L, Zhu L. Efficient defenses against output poisoning attacks on local differential privacy. IEEE Trans Inf Forensics Secur. 2023;18:5506–21. doi:10.1109/TIFS.2023.3305873.
84. Feng S, Yu H, Zhu Y. MMVFL: a simple vertical federated learning framework for multi-class multi-participant scenarios. Sensors. 2024;24(2):619. doi:10.3390/s24020619.
85. Wang S, Gai K, Yu J, Zhang Z, Zhu L. PraVFed: practical heterogeneous vertical federated learning via representation learning. IEEE Trans Inf Forensics Secur. 2025;20:2693–705. doi:10.1109/TIFS.2025.3530700.
86. Zhang R, Li H, Tian L, Hao M, Zhang Y. Vertical federated learning across heterogeneous regions for industry 4.0. IEEE Trans Ind Inform. 2024;20(8):10145–55. doi:10.1109/TII.2024.3393492.
87. Valdeira P, Xavier J, Soares C, Chi Y. Communication-efficient vertical federated learning via compressed error feedback. IEEE Trans Signal Process. 2025;73(276):1065–80. doi:10.1109/TSP.2025.3540655.
88. Fan K, Hong J, Li W, Zhao X, Li H, Yang Y. FLSG: a novel defense strategy against inference attacks in vertical federated learning. IEEE Internet Things J. 2024;11(2):1816–26. doi:10.1109/JIOT.2023.3302792.
89. Gong M, Zhang Y, Gao Y, Qin AK, Wu Y, Wang S, et al. A multi-modal vertical federated learning framework based on homomorphic encryption. IEEE Trans Inf Forensics Secur. 2024;19:1826–39. doi:10.1109/TIFS.2023.3340994.
90. Xu W, Zhu H, Zheng Y, Wang F, Zhao J, Liu Z, et al. ELXGB: an efficient and privacy-preserving XGBoost for vertical federated learning. IEEE Trans Serv Comput. 2024;17(3):878–92. doi:10.1109/TSC.2024.3394706.

91. Xiao Y, Li X, Li T, Wang R, Pang Y, Wang G. A distributed generative adversarial network for data augmentation under vertical federated learning. IEEE Trans Big Data. 2025;11(1):74–85. doi:10.1109/TBDATA.2024.3375150.
92. Hao W, El-Khamy M, Lee J, Zhang J, Liang KJ, Chen C, et al. Towards fair federated learning with zero-shot data augmentation. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Piscataway, NJ, USA: IEEE; 2021. p. 3305–14. doi:10.1109/CVPRW53098.2021.00369.
93. Yan Y, Wang H, Huang Y, He N, Zhu L, Xu Y, et al. Cross-modal vertical federated learning for MRI reconstruction. IEEE J Biomed Health Inform. 2024;28(11):6384–94. doi:10.1109/JBHI.2024.3360720.
94. Wang F, Hugh E, Li B. More than enough is too much: adaptive defenses against gradient leakage in production federated learning. IEEE/ACM Trans Netw. 2024;32(4):3061–75. doi:10.1109/TNET.2024.3377655.
95. Hu J, Wang Z, Shen Y, Lin B, Sun P, Pang X, et al. Shield against gradient leakage attacks: adaptive privacy-preserving federated learning. IEEE/ACM Trans Netw. 2024;32(2):1407–22. doi:10.1109/TNET.2023.3317870.
96. Yu C, Shen S, Wang S, Zhang K, Zhao H. Communication-efficient hybrid federated learning for e-health with horizontal and vertical data partitioning. IEEE Trans Neural Netw Learn Syst. 2025;36(3):5614–28. doi:10.1109/TNNLS.2024.3383748.
97. Peng Y, Wu Y, Bian J, Xu J. Hybrid federated learning for multimodal IoT systems. IEEE Internet Things J. 2024;11(21):34055–64. doi:10.1109/JIOT.2024.3443267.
98. Yi L, Wang G, Liu X, Shi Z, Yu H. FedGH: heterogeneous federated learning with generalized global header. In: Proceedings of the 31st ACM International Conference on Multimedia; 2023 Oct 29–Nov 3; Ottawa, ON, Canada. New York, NY, USA: ACM; 2023. p. 8686–96. doi:10.1145/3581783.3611781.
99. Banerjee S, Bhuyan D, Elmroth E, Bhuyan M. Cost-efficient feature selection for horizontal federated learning. IEEE Trans Artif Intell. 2024;5(12):6551–65. doi:10.1109/TAI.2024.3436664.
100. Dai Q, Yan T, Ren P. FedCSR: a new cluster sampling based on rotation mechanism in horizontal federated learning. Comput Commun. 2023;210(5):312–20. doi:10.1016/j.comcom.2023.08.016.
101. Zhang L, Li A, Peng H, Han F, Huang F, Li XY. Privacy-preserving data selection for horizontal and vertical federated learning. IEEE Trans Parallel Distrib Syst. 2024;35(11):2054–68. doi:10.1109/TPDS.2024.3439709.
102. Qiu W, Quan C, Zhu L, Yu Y, Wang Z, Ma Y, et al. Heart sound abnormality detection from multi-institutional collaboration: introducing a federated learning framework. IEEE Trans Biomed Eng. 2024;71(10):2802–13. doi:10.1109/TBME.2024.3393557.
103. Pang Y, Ni Z, Zhong X. Federated learning for crowd counting in smart surveillance systems. IEEE Internet Things J. 2024;11(3):5200–9. doi:10.1109/JIOT.2023.3305933.
104. Noman AA, Rahaman M, Pranto TH, Rahman RM. Blockchain for medical collaboration: a federated learning-based approach for multi-class respiratory disease classification. Healthcare Analytics. 2023;3(5):100135. doi:10.1016/j.health.2023.100135.
105. Kandil S, Marzbani F, Shamayleh A. Beyond blockchain: Enhancing data sharing in supply chains through horizontal federated learning in IoT-enabled VMI systems. In: 2024 Advances in Science and Engineering Technology International Conferences (ASET); 2024 Feb 26–29; Abu Dhabi, United Arab Emirates. Piscataway, NJ, USA: IEEE; 2024. p. 1–7. doi:10.1109/ASET60340.2024.10708671.
106. He Y, Yang Y, Yao T, He W. Improved algorithm for image classification datasets using federated transfer learning. In: 2023 IEEE 14th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP); 2023 Nov 24–26; Beijing, China. Piscataway, NJ, USA: IEEE; 2023. p. 1–6. doi:10.1109/PAAP60200.2023.10391706.
107. Naseh D, Abdollahpour M, Tarchi D. Real-world implementation and performance analysis of distributed learning frameworks for 6G IoT applications. Information. 2024;15(4):190. doi:10.3390/info15040190.
108. Rajesh LT, Das T, Shukla RM, Sengupta S. Give and take: federated transfer learning for industrial IoT network intrusion detection. In: 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2023 Nov 1–3; Exeter, UK. Piscataway, NJ, USA: IEEE; 2023. p. 2365–71. doi:10.1109/TrustCom60117.2023.00333.
109. Shahnazeer CK, Sureshkumar G. Federated transfer learning for early detection of multi-organ failure: a scalable predictive healthcare framework. In: 2025 International Conference on Machine Learning and Autonomous

Systems (ICMLAS); 2025 Mar 10–12; Prawet, Thailand. Piscataway, NJ, USA: IEEE; 2025. p. 19–24. doi:10.1109/ICMLAS64557.2025.10968928.
110. Zhao H, Pan Z, Wang Y, Ying Z, Xu L, Tan YA. Personalized label inference attack in federated transfer learning via contrastive meta learning. Proc AAAI Conf Artif Intell. 2025;39(21):22777–85. doi:10.1609/aaai.v39i21.34438.
111. Wang T, Dong ZY, Su L. Blockchain-enabled federated transfer learning for anomaly detection of power lines. In: 2024 IEEE Power & Energy Society General Meeting (PESGM); 2024 Jul 21–25; Seattle, WA, USA. Piscataway, NJ, USA: IEEE; 2024. p. 1–5. doi:10.1109/PESGM51994.2024.10689083.
112. Wan L, Ning J, Li Y, Li C, Li K. Intelligent fault diagnosis via ring-based decentralized federated transfer learning. Knowl Based Syst. 2024;284(5):111288. doi:10.1016/j.knosys.2023.111288.
113. Wang T, Ren C, Dong ZY, Yip C. Domain-adaptive clustered federated transfer learning for EV charging demand forecasting. IEEE Trans Power Syst. 2025;40(2):1241–54. doi:10.1109/TPWRS.2024.3449339.
114. Campos EM, Vidal AG, Hernández Ramos JL, Skarmeta A. Federated transfer learning for energy efficiency in smart buildings. In: IEEE INFOCOM, 2023—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 2023 May 17–20; Hoboken, NJ, USA. Piscataway, NJ, USA: IEEE; 2023. p. 1–6. doi:10.1109/INFOCOMWKSHPS57453.2023.10225844.
115. Jin H, Bai D, Yao D, Dai Y, Gu L, Yu C, et al. Personalized edge intelligence via federated self-knowledge distillation. IEEE Trans Parallel Distrib Syst. 2023;34(2):567–80. doi:10.1109/TPDS.2022.3225185.
116. Chen D, Yao L, Gao D, Ding B, Li Y. Efficient personalized federated learning via sparse model-adaptation. In: Proceedings of the 40th International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. p. 5234–56.
117. Yi L, Shi X, Wang N, Wang G, Liu X, Shi Z, et al. pFedKT: personalized federated learning with dual knowledge transfer. Knowl Based Syst. 2024;292(1–2):111633. doi:10.1016/j.knosys.2024.111633.
118. Wu L, Guo S, Ding Y, Wang J, Xu W, Zhan Y, et al. Rethinking personalized client collaboration in federated learning. IEEE Trans Mob Comput. 2024;23(12):11227–39. doi:10.1109/TMC.2024.3396218.
119. Sabah F, Chen Y, Yang Z, Raheem A, Azam M, Ahmad N, et al. FairDPFL-SCS: fair dynamic personalized federated learning with strategic client selection for improved accuracy and fairness. Inf Fusion. 2025;115(3):102756. doi:10.1016/j.inffus.2024.102756.
120. Li T, Sanjabi M, Beirami A, Smith V. Fair resource allocation in federated learning. arXiv:1905.10497. 2020.
121. Zhao H, Liu Q, Sun H, Xu L, Zhang W, Zhao Y, et al. Community awareness personalized federated learning for defect detection. IEEE Trans Comput Soc Syst. 2024;11(6):8064–77. doi:10.1109/TCSS.2024.3405556.
122. Wang Q, Chen S, Wu M. Communication-efficient personalized federated learning with privacy-preserving. IEEE Trans Netw Serv Manage. 2024;21(2):2374–88. doi:10.1109/TNSM.2023.3323129.
123. Xiong A, Zhou H, Song Y, Wang D, Wei X, Li D, et al. A multi-task based clustering personalized federated learning method. Big Data Min Anal. 2024;7(4):1017–30. doi:10.26599/BDMA.2024.9020001.
124. Zhou X, Yang Q, Zheng X, Liang W, Wang KI, Ma J, et al. Personalized federated learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse. IEEE J Sel Areas Commun. 2024;42(4):817–31. doi:10.1109/JSAC.2023.3345431.
125. Huang CG, Li H, Peng W, Tang LC, Ye ZS. Personalized federated transfer learning for cycle-life prediction of lithium-ion batteries in heterogeneous clients with data privacy protection. IEEE Internet Things J. 2024;11(22):36895–906. doi:10.1109/JIOT.2024.3433460.
126. Yu H, Yang X, Gao X, Kang Y, Wang H, Zhang J, et al. Personalized federated continual learning via multi-granularity prompt. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024 Aug 25–29; Barcelona, Spain. New York, NY, USA: ACM; 2024. p. 4023–34. doi:10.1145/3637528.3671948.
127. Liu H, Li B, Gao C, Xie P, Zhao C. Privacy-encoded federated learning against gradient-based data reconstruction attacks. IEEE Trans Inf Forensics Secur. 2023;18(1):5860–75. doi:10.1109/TIFS.2023.3309095.
128. Hu J, Du J, Wang Z, Pang X, Zhou Y, Sun P, et al. Does differential privacy really protect federated learning from gradient leakage attacks? IEEE Trans Mob Comput. 2024;23(12):12635–49. doi:10.1109/TMC.2024.3417930.

129. Chang W, Zhu T. Gradient-based defense methods for data leakage in vertical federated learning. Comput Secur. 2024;139:103744. doi:10.1016/j.cose.2024.103744.
130. Lang N, Cohen A, Shlezinger N. Stragglers-aware low-latency synchronous federated learning via layer-wise model updates. IEEE Trans Commun. 2025;73(5):3333–46. doi:10.1109/TCOMM.2024.3486979.
131. Liu J, Zheng J, Zhang J, Xiang L, Ng DWK, Ge X. Delay-aware online resource allocation for buffer-aided synchronous federated learning over wireless networks. IEEE Access. 2024;12:164862–77. doi:10.1109/ACCESS.2024.3489657.
132. Rajesh S, Gakhar I, Shorey R, Verma R. Unveiling the trade-offs: a parameter-centric comparison of synchronous and asynchronous federated learning. In: 2025 17th International Conference on Communication Systems and Networks (COMSNETS); Piscataway, NJ, USA: IEEE; 2025. p. 1013–7. doi:10.1109/COMSNETS63942.2025.10885640.
133. Jmal H, Piamrat K, Aouedi O. TUNE-FL: adaptive semi-synchronous semi-decentralized federated learning. In: 2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC); 2025 Jan 10–13; Las Vegas, NV, USA. Piscataway, NJ, USA: IEEE; 2025. p. 1–6. doi:10.1109/CCNC54725.2025.10975902.
134. Yu L, Sun X, Albelaihi R, Park C, Shao S. Dynamic client clustering, bandwidth allocation, and workload optimization for semi-synchronous federated learning. Electronics. 2024;13(23):4585. doi:10.3390/electronics13234585.
135. Roy A, Mahanta DR, Mahanta LB. A semi-synchronous federated learning framework with chaos-based encryption for enhanced security in medical image sharing. Results Eng. 2025;25(2):103886. doi:10.1016/j.rineng.2024.103886.
136. Zhou H, Li M, Sun P, Guo B, Yu Z. Accelerating federated learning via parameter selection and pre-synchronization in mobile edge-cloud networks. IEEE Trans Mob Comput. 2024;23(11):10313–28. doi:10.1109/TMC.2024.3376636.
137. Jiang Z, Xu Y, Xu H, Wang Z, Liu J, Chen Q, et al. Computation and communication efficient federated learning with adaptive model pruning. IEEE Trans Mob Comput. 2024;23(3):2003–21. doi:10.1109/TMC.2023.3247798.
138. Miao Y, Liu Z, Li X, Li M, Li H, Choo KKR, et al. Robust asynchronous federated learning with time-weighted and stale model aggregation. IEEE Trans Dependable Secure Comput. 2024;21(4):2361–75. doi:10.1109/TDSC.2023.3304788.
139. Zhu H, Kuang J, Yang M, Qian H. Client selection with staleness compensation in asynchronous federated learning. IEEE Trans Veh Technol. 2023;72(3):4124–9. doi:10.1109/TVT.2022.3220809.
140. Hu CH, Chen Z, Larsson EG. Scheduling and aggregation design for asynchronous federated learning over wireless networks. IEEE J Sel Areas Commun. 2023;41(4):874–86. doi:10.1109/JSAC.2023.3242719.
141. Xu Y, Ma Z, Xu H, Chen S, Liu J, Xue Y. FedLC: accelerating asynchronous federated learning in edge computing. IEEE Trans Mob Comput. 2024;23(5):5327–43. doi:10.1109/TMC.2023.3307610.
142. Chen Z, Yi W, Shin H, Nallanathan A. Adaptive semi-asynchronous federated learning over wireless networks. IEEE Trans Commun. 2025;73(1):394–409. doi:10.1109/TCOMM.2024.3425635.
143. Sha X, Sun W, Liu X, Luo Y, Luo C. Enhancing edge-assisted federated learning with asynchronous aggregation and cluster pairing. Electronics. 2024;13(11):2135. doi:10.3390/electronics13112135.
144. Zhou X, Liang W, Kawai A, Fueda K, She J, Wang KI. Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems. IEEE Trans Intell Transp Syst. 2024;25(7):6658–66. doi:10.1109/TITS.2024.3362058.
145. Gauthier F, Gogineni VC, Werner S, Huang YF, Kuh A. Asynchronous online federated learning with reduced communication requirements. IEEE Internet Things J. 2023;10(23):20761–75. doi:10.1109/JIOT.2023.3314923.
146. Xie H, Xia M, Wu P, Wang S, Huang K. Decentralized federated learning with asynchronous parameter sharing for large-scale IoT networks. IEEE Internet Things J. 2024;11(21):34123–39. doi:10.1109/JIOT.2024.3354869.
147. Fang B, Zhang T. Deeper leakage from gradients through membership inference attack. In: 2024 7th International Conference on Information and Computer Technologies (ICICT); 2024 Mar 15–17; Honolulu, HI, USA. Piscataway, NJ, USA: IEEE; 2024. p. 295–300. doi:10.1109/ICICT62343.2024.00054.
148. Yang H, Xue D, Ge M, Li J, Xu G, Li H, et al. Fast generation-based gradient leakage attacks: an approach to generate training data directly from the gradient. IEEE Trans Dependable Secure Comput. 2025;22(1):132–45. doi:10.1109/TDSC.2024.3387570.

149. Geng J, Mou Y, Li Q, Li F, Beyan O, Decker S, et al. Improved gradient inversion attacks and defenses in federated learning. IEEE Trans Big Data. 2024;10(6):839–50. doi:10.1109/TBDATA.2023.3239116.
150. Madni HA, Umer RM, Foresti GL. Blockchain-based swarm learning for the mitigation of gradient leakage in federated learning. IEEE Access. 2023;11:16549–56. doi:10.1109/ACCESS.2023.3246126.
151. Chen S, Miao Y, Li X, Zhao C. Compressed-sensing-based practical and efficient privacy-preserving federated learning. IEEE Internet Things J. 2024;11(8):14017–30. doi:10.1109/JIOT.2023.3339495.
152. Chen Y, Abrahamyan L, Sahli H, Deligiannis N. Learned parameter compression for efficient and privacy-preserving federated learning. IEEE Open J Commun Soc. 2024;5:3503–16. doi:10.1109/OJCOMS.2024.3409191.
153. Xiao X, Tang Z, Li C, Jiang B, Li K. SBPA: sybil-based backdoor poisoning attacks for distributed big data in AIoT-based federated learning system. IEEE Trans Big Data. 2024;10(6):827–38. doi:10.1109/TBDATA.2022.3224392.
154. Mbow M, Takahashi T, Sakurai K. Poisoning attacks on federated-learning based NIDS. In: 2024 Twelfth International Symposium on Computing and Networking Workshops (CANDARW); 2024 Nov 26–29; Naha, Japan. Piscataway, NJ, USA: IEEE; 2024. p. 69–75. doi:10.1109/CANDARW64572.2024.00020.
155. Zhou W, Zhang D, Wang H, Li J, Jiang M. A meta-reinforcement learning-based poisoning attack framework against federated learning. IEEE Access. 2025;13(268):28628–44. doi:10.1109/ACCESS.2025.3538891.
156. Nowroozi E, Haider I, Taheri R, Conti M. Federated learning under attack: exposing vulnerabilities through data poisoning attacks in computer networks. IEEE Trans Netw Serv Manage. 2025;22(1):822–31. doi:10.1109/TNSM.2025.3525554.
157. Thein TT, Shiraishi Y, Morii M. Personalized federated learning-based intrusion detection system: poisoning attack and defense. Future Gener Comput Syst. 2024;153(3):182–92. doi:10.1016/j.future.2023.10.005.
158. Chen L, Liu X, Wang A, Zhai W, Cheng X. FLSAD: defending backdoor attacks in federated learning via self-attention distillation. Symmetry. 2024;16(11):1497. doi:10.3390/sym16111497.
159. Ozdayi MS, Kantarcioglu M, Gel YR. Defending against backdoors in federated learning with robust learning rate. Proc AAAI Conf Artif Intell. 2021;35(10):9268–76. doi:10.1609/aaai.v35i10.17118.
160. Awan S, Luo B, Li F. CONTRA: defending against poisoning attacks in federated learning. In: Computer Security—ESORICS 2021: 26th European Symposium on Research in Computer Security; 2021 Oct 4–8; Darmstadt, Germany. Berlin/Heidelberg, Germany: Springer-Verlag; 2021. p. 455–75. doi:10.1007/978-3-030-88418-5_22.
161. Miao Y, Yan X, Li X, Xu S, Liu X, Li H, et al. RFed: robustness-enhanced privacy-preserving federated learning against poisoning attack. IEEE Trans Inf Forensics Secur. 2024;19:5814–27. doi:10.1109/TIFS.2024.3402113.
162. Li X, Wen M, He S, Lu R, Wang L. A privacy-preserving federated learning scheme against poisoning attacks in smart grid. IEEE Internet Things J. 2024;11(9):16805–16. doi:10.1109/JIOT.2024.3365142.
163. Shen L, Ke Z, Shi J, Zhang X, Sun Y, Zhao J, et al. SPEFL: efficient security and privacy-enhanced federated learning against poisoning attacks. IEEE Internet Things J. 2024;11(8):13437–51. doi:10.1109/JIOT.2023.3339638.
164. Yin C, Zeng Q. Defending against data poisoning attack in federated learning with non-IID data. IEEE Trans Comput Soc Syst. 2024;11(2):2313–25. doi:10.1109/TCSS.2023.3296885.
165. Li X, Yang X, Zhou Z, Lu R. Efficiently achieving privacy preservation and poisoning attack resistance in federated learning. IEEE Trans Inf Forensics Secur. 2024;19:4358–73. doi:10.1109/TIFS.2024.3378006.
166. So J, Güler B, Avestimehr AS. Byzantine-resilient secure federated learning. IEEE J Sel Areas Commun. 2021;39(7):2168–81. doi:10.1109/JSAC.2020.3041404.
167. Zhang Z, Wu L, Ma C, Li J, Wang J, Wang Q, et al. LSFL: a lightweight and secure federated learning scheme for edge computing. IEEE Trans Inf Forensics Secur. 2023;18:365–79. doi:10.1109/TIFS.2022.3221899.
168. Basak S, Chatterjee K. DPAD: data poisoning attack defense mechanism for federated learning-based system. Comput Electr Eng. 2025;121(1):109893. doi:10.1016/j.compeleceng.2024.109893.
169. Feng X, Cheng W, Cao C, Wang L, Sheng V. S. DPFLA: defending private federated learning against poisoning attacks. IEEE Trans Serv Comput. 2024;17(4):1480–91. doi:10.1109/TSC.2024.3376255.
170. Liu J, Li X, Liu X, Zhang H, Miao Y, Deng RH. DefendFL: a privacy-preserving federated learning scheme against poisoning attacks. IEEE Trans Neural Netw Learn Syst. 2025;36(5):9098–111. doi:10.1109/TNNLS.2024.3423397.
171. Chen X, Yu H, Jia X, Yu X. APFed: anti-poisoning attacks in privacy-preserving heterogeneous federated learning. IEEE Trans Inf Forensics Secur. 2023;18:5749–61. doi:10.1109/TIFS.2023.3315125.

172. Wang T, Huang N, Wu Y, Gao J, Quek TQS. Latency-oriented secure wireless federated learning: a channel-sharing approach with artificial jamming. IEEE Internet Things J. 2023;10(11):9675–89. doi:10.1109/JIOT.2023.3234422.
173. Kim G, Kim Y. The threat of disruptive jamming to blockchain-based decentralized federated learning in wireless networks. Sensors. 2024;24(2):535. doi:10.3390/s24020535.
174. Jeon SE, Lee SJ, Lee YR, Yu H, Lee IG. Machine learning-based cooperative clustering for detecting and mitigating jamming attacks in beyond 5G networks. Inf Syst Front. 2024;11(10):1. doi:10.1007/s10796-024-10534-6.
175. de Caldas Filho FL, Soares SCM, Oroski E, de Oliveira Albuquerque R, da Mata RZA, de Mendonça FLL, et al. Botnet detection and mitigation model for IoT networks using federated learning. Sensors. 2023;23(14):6305. doi:10.3390/s23146305.
176. Barkatsa S, Diamanti M, Charatsaris P, Voikos S, Tsiropoulou EE, Papavassiliou S. Coordinated jamming and poisoning attack detection and mitigation in wireless federated learning networks. IEEE Open J Commun Soc. 2025;6:3745–59. doi:10.1109/OJCOMS.2025.3558672.
177. Meftah A, Do TN, Kaddoum G, Talhi C. Federated learning-enabled jamming detection for stochastic terrestrial and non-terrestrial networks. IEEE Trans Green Commun Netw. 2025;9(1):271–90. doi:10.1109/TGCN.2024.3425792.
178. Houda ZAE, Naboulsi D, Kaddoum G. A privacy-preserving collaborative jamming attacks detection framework using federated learning. IEEE Internet Things J. 2024;11(7):12153–64. doi:10.1109/JIOT.2023.3333870.
179. Li C, He A, Liu G, Wen Y, Chronopoulos AT, Giannakos A. RFL-APIA: a comprehensive framework for mitigating poisoning attacks and promoting model aggregation in IIoT federated learning. IEEE Trans Ind Informat. 2024;20(11):12935–44. doi:10.1109/TII.2024.3431020.
180. Ben Saad S, Brik B, Ksentini A. Toward securing federated learning against poisoning attacks in zero touch B5G networks. IEEE Trans Netw Serv Manag. 2023;20(2):1612–24. doi:10.1109/TNSM.2023.3278838.
181. Hossain MA, Islam MS. Towards decentralized cybersecurity: a novel privacy-preserving federated learning approach for botnet attack detection. Blockchain Res Appl. 2025;24(3):100355. doi:10.1016/j.bcra.2025.100355.
182. Elleuch I, Pourranjbar A, Kaddoum G. Deep cross-check Q-learning for jamming mitigation in wireless networks. IEEE Wireless Commun Lett. 2024;13(5):1448–52. doi:10.1109/LWC.2024.3374215.
183. Barkatsa S, Diamanti M, Charatsaris P, Tsiropoulou EE, Papavassiliou S. Jamming attack mitigation in wireless federated learning networks using Bayesian games. IEEE Netw Lett. 2024;6(4):247–51. doi:10.1109/LNET.2024.3499360.
184. Houda ZAE, Moudoud H, Brik B. Federated deep reinforcement learning for efficient jamming attack mitigation in O-RAN. IEEE Trans Veh Technol. 2024;73(7):9334–43. doi:10.1109/TVT.2024.3359998.