

Federated Learning: A Survey of Recent Advances, Challenges and Applications

Shreya Regish*, Riya K Prakash**, Dr. L.C Manikandan***

*B-Tech. Student, Department of CSE, Universal Engineering College, Kerala, India.

**Assistant Professor, Department of CSE, Universal Engineering College, Kerala, India.

***Professor, Department of CSE, Universal Engineering College, Kerala, India.

E-mail: shreyaxr36@gmail.com, riyajothish13@gmail.com, lcmanikandan@gmail.com

Abstract:

Federated Learning has emerged as a transformative paradigm which enables many devices or organizations work together to train an AI model without sharing their private data. This survey provides a comprehensive examination of Federated Learning, where it systematically reviews advances in architectures, aggregation algorithms, privacy-preserving techniques, and security mechanisms. The survey also explains how Federated Learning has developed from basic methods like Federated Averaging (FedAvg) to more advanced sophisticated frameworks like addressing data heterogeneity, communication efficiency and personalization that can handle problems such as Different types of data on different devices (statistical heterogeneity), Devices with different processing power and internet speed (system heterogeneity), High communication costs and the need to create customized models for different users. Emerging paradigms including decentralized Federated Learning, clustered Federated Learning, personalized Federated Learning, and federated prompt learning are explored. Finally, the survey explores the remaining challenges and future possibilities. It highlights the need for Federated Learning systems that are more adaptive, efficient, secure, and trustworthy for real-world applications such as healthcare, IoT devices, edge computing, and smart systems.

Keywords — Federated Learning, Privacy Preservation, Data Heterogeneity, Communication Efficiency, Secure Aggregation, Personalization, Decentralized Architecture, Large Language Models.

I. INTRODUCTION

Federated Learning was introduced by researchers at Google in 2016. The key idea was: “Bring the model to the data, instead of bringing the data to the model.” The approach was designed especially for mobile devices, where data such as typing information stays on the user’s phone. Google published the influential paper “Communication-

Efficient Learning of Deep Networks from Decentralized Data.” It formally described the Federated Averaging (FedAvg) algorithm. The main feature of federated learning is to ensure the privacy of users, but it is very different from traditional privacy protection algorithms applied in the field of big data, such as differential privacy and k-order

anonymity. Federated learning mainly protects user privacy by exchanging encrypted processed parameters, while the attackers cannot get source data. All these guarantee that federated learning will not leak user privacy at the data level, and there is no violation of GDPR and other bills. Federated learning can be divided into horizontal federated learning, vertical federated learning and federated transfer learning according to the distribution of data. Horizontal federated learning is suitable in the case that the user features of the two datasets overlap a lot, but the users overlap little. Vertical federated learning is available in the case that the user features of the two datasets overlap little, but the users

overlap a lot. In the case that the users and user features of the two datasets both rarely overlap, we can use transfer learning to overcome the lack of data or tags [2]. As FL research has expanded, several limitations of a single global model have become clearer. Non-IID data can cause local optimization directions to diverge, while device and system differences introduce stragglers, synchronization

delays, and unequal computational burdens. Communication can dominate training cost, particularly when models are large or devices have constrained networks. At the same time, model updates and intermediate representations can themselves reveal sensitive information, so keeping raw data local does not automatically provide complete privacy [1], [2], [13]. Recent literature therefore broadens the objective of FL beyond simply producing one global model. Clustered FL groups clients with similar data distributions to produce specialized models; decentralized FL removes the central coordinator and uses peer-to-peer coordination; VFL addresses collaboration when useful features are distributed across organizations; and recent FedLLM and FedFM studies investigate how federated learning can adapt large language and foundation models while controlling their substantial computational and communication demands [5], [6], [8], [9], [11], [16]. Emerging paradigms including decentralized Federated Learning, clustered Federated Learning, personalized Federated Learning, and federated prompt learning are explored.

Finally, the survey explores the remaining challenges and future possibilities. It highlights the need for Federated Learning systems that are more adaptive, efficient, secure, and trustworthy for real-world applications such as healthcare, IoT devices, edge computing, and smart systems.

A. Scope and Contributions

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. We provide a structured review organized around three core challenges: data

distribution, model synchronization, and defense against attack [13]. Our specific contributions are:

- A unified taxonomy categorizing FL approaches by architecture, data distribution, and training methodology
- Comprehensive analysis of aggregation algorithms addressing statistical and system heterogeneity
- Systematic review of privacy-preserving mechanisms and security defenses
- Examination of emerging paradigms including hierarchical, personalized, and blockchain-based FL
- Identification of open challenges and future research directions. This survey structures the selected papers around these developments. Rather than treating each paper as an isolated contribution, it connects their recurring themes: privacy preservation, communication and computation on security, specialized FL architectures, and practical applications. The aim is to provide a coherent view of how the field is evolving and where the literature identifies remaining gaps.

II. FEDERATED LEARNING FOUNDATIONS AND DATA DISTRIBUTION

A. Basic Workflow

The basic workflow of Federated Learning consists of several steps. First, a coordinating server initializes a global machine learning model and selects participating clients. The model is then sent to these clients, where each client trains it locally using its own dataset. The raw data remains on the client device and is not shared with the server. After local training, each client sends the updated model parameters to the coordinating server.

The server then aggregates the updates from multiple clients to create an improved global model. This updated model is sent back to the clients, and the process is repeated for several rounds until the model achieves the desired performance. [2], [5], [7]. Although Federated Learning keeps the original data on the participants' devices, this does not guarantee complete privacy. The information shared during training, such as model parameters, gradients, and other model updates, may still reveal sensitive information. Attackers could potentially use this information to obtain details about the original data. Therefore, additional privacy and security methods are needed to protect the information exchanged

during the training process [3], [9], [13], [16]. The following figure shows the basic workflow:

B. Basic Workflow

The basic workflow of Federated Learning consists of several steps. First, a coordinating server initializes a global machine learning model and selects participating clients. The model is then sent to these clients, where each client trains it locally using its own dataset. The raw data remains on the client device and is not shared with the server. After local training, each client sends the updated model parameters to the coordinating server. The server then aggregates the updates from multiple clients to create an improved global model. This updated model is sent back to the clients, and the process is repeated for several rounds until the model achieves the desired performance [2], [5], [7]. Although Federated Learning keeps the original data on the participants' devices, this does not guarantee complete privacy. The information shared during training, such as model parameters, gradients, and other model updates, may still reveal sensitive information. Attackers could potentially use this information to obtain details about the original data. Therefore, additional privacy and security methods are needed to protect the information exchanged during the training process [3], [9], [13], [16]. The following Fig.1 shows the basic workflow:

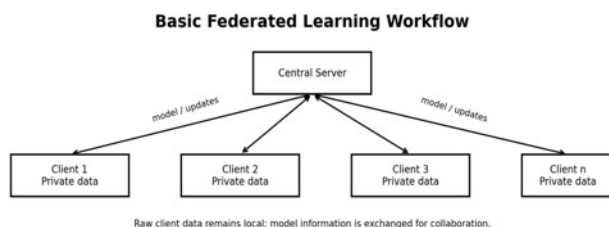


Fig.1 Redrawn schematic of the basic FL workflow

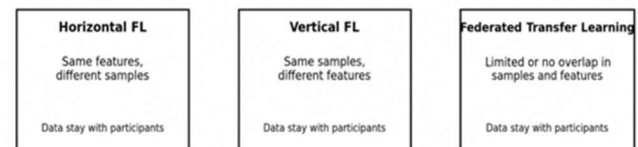
C. Data Distribution

Types of Federated Learning: Federated Learning can mainly be classified into Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL) based on how the data is distributed among participants [2], [16]. In HFL, participants have similar types of features but different data samples. This approach is commonly used in applications such as mobile

device learning. In VFL, participants have different features for the same or similar users, allowing them to combine complementary information without directly sharing their data.

FTL is used when participants have limited overlap in both their samples and features, and knowledge is transferred between them to support collaborative learning [2], [16]. Each type of Federated Learning has different challenges. HFL mainly faces issues such as non-IID data and differences between devices. VFL additionally needs to handle user or sample alignment and protect intermediate information that may reveal sensitive data. FTL faces challenges related to limited data overlap, knowledge transfer, and alignment between participants [2], [13], [14], [16]. The following Fig.2 shows the three data partition settings:

Three Data-Partition Settings in Federated Learning



Classification follows how samples and feature spaces are distributed across participants.

Fig. 2. Redrawn classification of HFL, VFL and FTL based on the data partition descriptions

III. RECENT ADVANCES IN FEDERATED LEARNING

A. Optimization and Model Training

FedAvg and FedSGD are the basic methods used for training models in Federated Learning. In FedSGD, clients calculate model updates using their local data and send them to the server. In FedAvg,

clients perform several local training steps and the server combines their model updates to create a new global model. Later methods such as Fed Prox, SCAFFOLD, and FedNova were developed to handle differences between clients and improve the training process [1], [3], [13]. These methods are important because clients may have different amounts and types of data, different devices, and different training conditions. Another important development is personalized Federated Learning.

A single global model may not work equally well for every client because their data and requirements can be different. Personalized FL allows models to

be adapted to individual clients or groups of similar clients [1], [5], [7], [13]. This can improve the performance of the model for specific users or environments. Personalization is also useful for foundation models and language models, where different users may need different models or responses [8], [10], [11].

B. Communication and Computation Efficiency

Communication is an important challenge in Federated Learning because clients and the server need to exchange information repeatedly. Sending large model updates many times can increase network usage and training time. To reduce communication costs, researchers use methods such as client selection, data compression, sparsification, quantization, and asynchronous communication [1], [2], [13]. Client selection reduces the number of participants in each round, while compression and quantization reduce the size of the information being transmitted. These methods help make FL more practical for devices with limited network resources. This challenge becomes greater when using large models such as Large Language Models (LLMs) because they contain a very large number of parameters. Training and exchanging the complete model can therefore require significant computing power, memory, and communication. Recent research explores techniques such as model compression, quantization, sparsification, and parameter-efficient fine-tuning to reduce the training cost [8], [9], [11]. Instead of updating the entire model, only a smaller part of the model can be trained and shared. This makes it easier to use large models in federated environments with limited resources.

C. Synchronization and System Heterogeneity

Federated Learning clients may have different levels of computing power, memory, internet connection, and availability. These differences are known as system or resource heterogeneity. For example, one mobile phone may have a fast processor and good internet connection, while another may have limited resources. In synchronous FL, all selected clients are generally expected to complete their local training before the server

continues. Therefore, slower clients can delay the entire training process. These slower clients are called stragglers. Asynchronous FL allows clients to send updates at different times, which can reduce waiting time, but it can also result in older or outdated updates [1], [13]. Therefore, synchronization is an important part of practical

Federated Learning. The goal is to balance training speed, model performance, and the available resources of different clients. Communication, computation, and privacy are also connected. For example, compression reduces the size of updates, client selection controls which devices participate, and asynchronous training affects when updates are received [13]. Together, these developments help Federated Learning work more efficiently in real-world environments with many different types of clients.

IV. PRIVACY AND SECURITY IN FEDERATED LEARNING

A. Privacy-Preserving Methods

Privacy is one of the main reasons for using Federated Learning. However, keeping data on local devices does not provide complete privacy. Additional methods are needed to protect the information share during training. Differential Privacy adds controlled noise to reduce the chance of revealing private information. Homomorphic Encryption allows calculations to be performed on encrypted data. Secure Multi-Party Computation and Secret Sharing allow participants to work together without directly revealing their private information. Secure Aggregation allows the server to receive the combined updates without seeing each client's individual update [1], [2], [3], [16]. These privacy methods can affect the performance of the system. Stronger security methods may require more computing power and communication. This is especially important in Vertical Federated Learning (VFL), where protecting intermediate information and communication between participants is also necessary [16].

B. Inference and Data Leakage Attacks

Federated Learning can still face attacks that try to obtain private information. Label inference attacks

attempt to find the labels or answers used by another participant. Feature inference attacks try to recover private features from shared information. Other attacks include gradient inversion, model inversion, and reconstruction attacks, which may use model updates to recover information about the original data [16]. Other common threats in FL include gradient leakage, membership inference, poisoning, and backdoor attacks [3], [7], [9], [13]. Membership inference tries to find out whether a particular person's data was used for training. Reconstruction and inversion attacks try to recover information from model updates. In poisoning attacks, an attacker sends harmful or incorrect updates to affect the global model. A backdoor attack makes the model behave normally in most cases but produce an attacker-controlled result for certain inputs.

C. Defense Strategies

Researchers have developed different methods to protect Federated Learning systems. These include secure aggregation, differential privacy, homomorphic encryption, secure multi-party computation, secret sharing, trusted execution environments, anomaly detection, and robust aggregation [3], [9], [11], [13], [16]. No single security method can protect against every type of attack. For example, encryption can protect information during communication but may increase computing requirements. Similarly, adding privacy noise can reduce information leakage but may also reduce the accuracy of the model. Recent research on Foundation Models and Large Language Models (LLMs) also considers new security problems. These include protecting model ownership, preventing model theft, and stopping malicious participants from changing the shared model [9], [11]. Therefore, security in Federated Learning needs to be considered throughout the complete process, from training and model updates to deployment and use.

V. HETEROGENEITY, PERSONALIZATION AND ADAPTATION

A. Statistical and Data Heterogeneity

One of the main challenges in Federated Learning is that clients usually have different types of data. This is called non-IID (non-independent and identically

distributed) data. For example, different users may have different preferences, classes, topics, or amounts of data. Because of these differences, the model updates from different clients may not be similar. As a result, one global model may not perform equally well for all clients [1], [2], [5], [13]. Several methods have been developed to handle this problem. FedProx and other optimization methods help reduce differences between client updates. Personalized FL allows each client to adapt the model to its own data. Client selection chooses suitable clients for training, while clustering groups clients with similar data. Knowledge distillation and transfer learning can also help clients share useful knowledge without directly combining all their data or model parameters [3], [5], [8], [11].

B. Model and Resource Heterogeneity

Clients can differ not only in their data but also in their hardware and resources. Some devices may have more memory, faster processors, better internet connections, or more battery power than others. These differences are called system or resource heterogeneity [1], [8], [9], [13]. These differences can make training difficult because some clients may take longer to complete their training. Large models such as LLMs make this problem even more serious because they require more memory and computing power. Therefore, methods such as parameter-efficient fine-tuning, model compression, and lightweight adapters are used to make large models easier to train in federated environments [8], [9], [11].

1. Personalization: In traditional Federated Learning, the goal is usually to create one global model for all clients. However, different clients may have different needs and data. Personalized Federated Learning allows the model to be adjusted for individual clients while still benefiting from collaborative training [1], [5], [7]. Personalization can be achieved through local adaptation, client clustering, personalized models, and prototype-based methods. In clustered FL, clients with similar data are grouped together and can receive a model that is better suited to their group [1], [5], [7]. Personalization is also becoming important for LLMs and foundation models. Different users or organizations may need different behaviours from

the same pretrained model. Techniques such as client specific fine tuning, heterogeneous adapters, and prompt learning allow clients to customize the model while sharing the main knowledge of the pretrained model [8], [10], [11].

VI. SPECIALIZED FEDERATED LEARNING PARADIGMS

A. Clustered Federated Learning (CFL)

- The Problem: Standard Federated Learning (FL) tries to force every device to share one single global model. But if users have wildly different data (e.g., users in different countries or climate zones), one model can't fit everyone well.
- The Solution: CFL groups (clusters) users with similar data together. Instead of making one global model for everyone, the system trains a few tailored models—one for each group.
- Distinctions Challenges:
- Core CFL vs. Clustered "X" FL: Core CFL groups users to train different models. Other approaches group users just to make network speeds faster or schedule devices better while still building one model.
- Key Challenges: Figure out how many groups to make, handle users whose habits change over time, and keep user data private while grouping them.[5]
- Where It's Used: IoT devices, smart grids, self-driving tech, and healthcare where user differences matter.

B. Decentralized Federated Learning (DFL)

- The Problem: Standard FL relies on one central server to coordinate everything. If that server crashes, gets hacked, or can't be trusted, the whole system fails.
- The Solution: DFL gets rid of the main server entirely. Devices talk directly to their neighbouring devices in a peer-to-peer (P2P) network or use blockchain technology to keep track of updates.
- Trade-offs:
- Eliminates the single point of failure.
- Makes network traffic heavier between devices, creates new privacy risks among peers, and makes it harder to detect malicious users.[6]

C. Vertical Federated Learning (VFL)

- The Problem (Horizontal vs. Vertical):

- Horizontal FL (Standard): Different people, same type of data (e.g., two hospitals with different patients, but recording the exact same lab tests).
- Vertical FL: Same people, different types of data (e.g., a bank and an e-commerce platform that serve the same customers, but hold totally different details about them).
- The Solution: VFL lets companies combine their complementary features for shared customers to train a smarter model together—without ever sharing their private customer data directly with each other.[14],[16]

VII. VERTICAL FL: SECURITY AND PRACTICAL DEPLOYMENT

A. Privacy-preserving VFL protocols

The VFL literature provides a particularly detailed hierarchy of privacy-preserving protocols. A basic protocol keeps private data and models local while allowing intermediate results to be transmitted. A standard protocol additionally protects transmitted intermediate results, for example through homomorphic encryption. An enhanced protocol protects the broader training procedure using secure multi-party computation, while a strict protocol additionally protects learned models and reveals only the required inference results [16]. This progression demonstrates the central privacy–efficiency trade-off. Protecting more information generally requires more computation or communication.

The VFL survey describes cryptographic defenses based on garbled circuits, secret sharing homomorphic encryption, functional encryption, and trusted execution environments, with different schemes targeting linear models, neural networks, and tree-based models [16].

B. VFL inference threats

VFL creates two directions of inference risk. A passive party may attempt to infer labels owned by an active party, while an active party or other adversary may attempt to recover features held by passive parties. The surveyed literature includes direct label inference, gradient inversion, model completion, model inversion, reverse multiplication, reverse sum strategies, and other attacks tailored to

the model and protocol [16]. These attacks demonstrate why the phrase “raw data is never shared” is insufficient as a complete privacy argument. Intermediate embeddings, gradients, model outputs, and other protocol artifacts can become information channels. The VFL literature therefore emphasizes matching the privacy protocol to the threat model and the information that each party is permitted to observe [14], [16].

C. Practical applicability

Real-world deployment introduces difficulties that may not be visible in controlled experimental settings. The practical VFL survey reports substantial mismatch between common research assumptions and observed real-world data distributions. It highlights situations with insufficient feature overlap, incomplete record matching, and imbalanced feature distributions, and organizes practical VFL research around key alignment, feature balance, communication, and trustworthiness [15]. This finding connects with the broader surveys: practical FL needs more than a strong learning algorithm. It requires realistic assumptions about data availability, communication, resource constraints, security, and participant behaviour. VFL research therefore increasingly focuses on limited labels, sparse overlap, communication-efficient training, client asynchrony, and deployment-oriented evaluation [14], [15], [16].

VIII. FEDERATED LARGE LANGUAGE MODELS AND PROMPT LEARNING

A. Motivation for federated LLMs

Large Language Models have created a new FL setting because their training requires large amounts of data and substantial computational resources, while useful domain specific data may be private and distributed. The FedLLM surveys describe healthcare, finance, legal services, recommendation, and other domains where centralized collection of sensitive text can be problematic [8], [9]. Federated learning provides a mechanism for collaborative adaptation without directly pooling the underlying local datasets. At the same time, LLM scale intensifies traditional FL problems. Large parameter counts increase communication and computation

costs, heterogeneous client resources make synchronization difficult, and model updates can expose information. The selected FedLLM literature therefore organizes recent work around efficiency, heterogeneity, personalization, privacy, security, and robustness [8], [9].

B. Federated fine-tuning and PEFT

Federated fine-tuning is a major direction in FedLLM research. The literature considers full-parameter finetuning as well as parameter-efficient approaches such as adapters, Lora, and prompt tuning. The central motivation is to reduce the number of trainable parameters and therefore Reduce computation, memory use, and communication [8],[9], [11]. Heterogeneous LoRA and adapter approaches are also discussed because clients may have different resource budgets or model-training configurations. Methods described in the selected papers include homogeneous adapters for aggregation, heterogeneous LoRA designs, different LoRA ranks, and aggregation strategies intended to preserve personalized representations [8].

C. Federated Prompt Learning

Federated Prompt Learning combines federated optimization with prompt-based adaptation. The uploaded FPL survey frames its analysis around motivations and enabling technologies, performance and efficiency trade-offs, and remaining security, privacy, robustness, and system challenges [10]. Instead of updating an entire foundation model, prompt learning can restrict collaboration to lightweight trainable components. The FPL literature covers the model lifecycle from pretraining and fine-tuning to applications, while considering personalization, heterogeneity, scalability, and communication. It also examines attacks and defenses relevant to federated prompt learning. This makes prompt-based adaptation a significant bridge between the resource constraints of FL and the scale of modern foundation models [10].

IX. FOUNDATION MODELS, APPLICATIONS AND COMPARATIVE SYNTHESIS

A. Federated Foundation Models

The Foundation Model survey presents FedFM as the combination of large pretrained models with federated learning for collaborative adaptation. Its organizing framework uses three major dimensions: efficiency, adaptability, and trustworthiness [11]. Efficiency includes parameter-efficient finetuning, compression, and zeroth-order optimization. Adaptability addresses data and resource heterogeneity, personalization, and domain adaptation. Trustworthiness covers privacy, security, intellectual property, and poisoning. The survey also notes that foundation models span language, vision, speech, and multimodal applications. Their general-purpose representations can be adapted to downstream tasks, but domain-specific adaptation may require proprietary data. Federated learning provides a way to use distributed private data while avoiding centralized data collection [11].

B. Applications

The selected papers report or review FL applications in healthcare, finance, recommendation systems, mobile and IoT environments, industrial systems, smart systems, and language technologies [1], [3], [4], [7], [11], [12]. Healthcare is repeatedly emphasized because medical data is sensitive and distributed across institutions. The medical-imaging review focuses specifically on federated approaches for cancerrelated imaging and covers lung, breast, brain, skin, and colorectal cancers [12]. The medical-imaging literature reports the use of personalization, heterogeneity, scalability, and communication. It also examines attacks and defenses relevant to federated prompt learning. This makes prompt-based adaptation a significant bridge between the resource constraints of FL and the scale of modern foundation models [10].

B. Applications

The selected papers report or review FL applications in healthcare, finance, recommendation systems, mobile and IoT environments, industrial systems, smart systems, and language technologies [1], [3], [4], [7], [11], [12]. Healthcare is repeatedly emphasized because medical data is sensitive and distributed across institutions. The medical-imaging review focuses specifically on federated approaches

for cancerrelated imaging and covers lung, breast, brain, skin, and colorectal cancers [12]. The medical-imaging literature reports the use of deep-learning architectures including UNet variants and several CNN-based models, while also identifying privacy frameworks such as homomorphic encryption and blockchain-backed mechanisms. It highlights data heterogeneity, limited datasets, communication costs, privacy risks, and the lack of sufficient clinical deployment benchmarks as important barriers [12]. Industrial and financial applications also benefit from complementary data held by different organizations. VFL is particularly relevant where institutions possess different features about the same users, while HFL is suitable when organizations or devices hold different samples with comparable feature spaces [2], [4], [14], [16]. Recommendation and language applications are increasingly connected with foundation models, creating a need for efficient and privacy-aware adaptation [8], [11].

X. CHALLENGES, FUTURE DIRECTIONS AND CONCLUSION**A. Open challenges identified across the selected papers**

- Privacy is still an active research problem because model updates, gradients, embeddings, and intermediate results can leak information even when raw data remains local [3], [13], [16].
- Heterogeneity remains multi-dimensional: data distributions, model structures, device resources, communication conditions, and task requirements can all differ [1], [8], [9], [13].
- Communication and computation costs continue to limit practical deployment, with the problem becoming more severe for foundation models and LLMs [2], [8], [11].
- Personalization requires a balance between local specialization and collaborative knowledge sharing. Clustered and personalized FL offer useful directions but introduce additional clustering, scalability, and privacy questions [5], [7].

B. Future directions

Across the selected papers, several future directions recur. First, FL systems need adaptive privacy mechanisms that consider the actual information exposed by different protocols. Second,

communication and computation should be jointly optimized with model quality rather than treated as independent objectives. Third, personalization and clustering should become more scalable and adaptive to changing client distributions. Fourth, decentralized systems require topology aware security models and incentive mechanisms that remain robust to manipulation [5], [6], [13].

For VFL, the literature points toward better handling of limited alignment, communication burden, client asynchrony, and stronger protection of embeddings, gradients, labels, and learned models [14], [15], [16]. For foundation models and LLMs, promising directions include more efficient parameter-efficient adaptation, federated pre-training, federated AI agents, few-shot learning, machine unlearning, and intellectual-property protection [8], [9], [10], [11]. Applications also need stronger evidence beyond benchmark datasets. The medical-imaging review specifically identifies a lack of clinical deployment benchmarks, while the practical VFL study shows that real data distributions can differ substantially from assumptions made in existing algorithms [12], [15]. Thus, future progress depends not only on new algorithms but also on realistic data, system, security, and evaluation settings.

C. Conclusions

The selected literature shows a clear progression in federated learning: from basic collaborative model aggregation toward specialized systems designed around heterogeneity, privacy, efficiency, personalization, and application requirements. HFL, VFL and FTL provide different foundations according to data distribution, while Clustered FL and Personalized FL address differences among clients. Decentralized FL changes the coordination model itself. VFL adds detailed privacy and inference challenges because different parties contribute complementary features. More recently, FedLLM, Federated Prompt Learning and FedFM extend the paradigm to large pretrained models, where parameter scale makes efficiency and trustworthiness even more important [5], [6], [8], [10], [11], [16]. Overall, the papers do not suggest that federated learning has eliminated the privacy, efficiency, or deployment problems associated with

machine learning. Instead, they show that the research focus is shifting toward managing those problems explicitly and jointly. The most important future goal is therefore a federated system that can adapt to heterogeneous participants, protect sensitive information, remain computationally and communicatively practical, withstand attacks, and demonstrate reliable performance under realistic application conditions.

ACKNOWLEDGMENT

The author acknowledges the guidance and support of Prof. Riya K Prakash and Dr.L.C.Manikandan during the preparation of this study.

REFERENCES

- [1] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics* 14 (2023) 513–535.
- [2] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, "A survey on federated learning," *Knowledge-Based Systems* 216 (2021) 106775.
- [3] A. A. Micheal, "A Comprehensive Survey of Federated Learning Approaches for Privacy-Preserving Machine Learning," *Tech-Sphere Journal of Pure and Applied Sciences* 2 (1) (2025).
- [4] L. Li, Y. Fan, M. Tse, K.-Y. Lin, "A review of applications in federated learning," *Computers Industrial Engineering* 149 (2020) 106854.
- [5] M. Ben Ali, O. El-Rifai, I. Megdiche, A. Peninou, O. Teste, "A survey on Clustered Federated Learning: Taxonomy, Analysis and Applications," *arXiv:2501.17512* (revised through June 2025).
- [6] E. Gabrielli, A. Di Pietro, D. Fenoglio, G. Pica, G. Tolomei, "A Survey on Decentralized Federated Learning," *survey manuscript*, 2026.
- [7] P. M. Mammen, "Federated Learning: Opportunities and Challenges," 2021.
- [8] Y. Yao, J. Zhang, J. Wu, C. Huang, Y. Xia, T. Yu, R. Zhang, S. Kim, R. Rossi, A. Li, L. Yao, J. McAuley, Y. Chen, C. Joe-Wong, "Federated Large Language Models: Current Progress and Future Directions", v3 (2026).
- [9] W. Jiang, Y. Luo, G. Deng, S. Chen, X. Yang, S. Wu, X. Gao, L. Liu, S. Fu, "Federated Large Language Models: Feasibility, Robustness, Security and Future Directions," 2025.
- [10] Q. Yang, C. Qiu, H. Zhang, P. Li, Y. Liu, Z. Tian, "Federated Prompt Learning: A Unified Framework,

- Empirical Analysis, and Future Directions,” manuscript submitted to ACM, 2026.
- [11] S. Li, F. Ye, M. Fang, J. Zhao, Y.-H. Chan, E. C. H. Ngai, T. Voigt, “Synergizing Foundation Models and Federated Learning: A Survey,” arXiv:2406.12844 (2024).
- [12] D. Ghosh, M. Mehjabin, M. E. Rayed, M. F. Mridha, M. M. Kabir, “Advancements and challenges of federated learning in medical imaging: a systematic literature review,” *Artificial Intelligence Review* 59 (2026) 87.
- [13] F. Mahmud, F. Mahmud, R. M. Rahman, “A Survey of Federated Learning: Advances in Architecture, Synchronization, and Security Threats,” *Computers, Materials Continua* 86 (3) (2026).
- [14] M. Ye, W. Shen, B. Du, E. Snezhko, V. Kovalev, P. C. Yuen, “Vertical Federated Learning for Effectiveness, Security, Applicability: A Survey,” June 2024, 40 pages, arXiv:2405.17495
- [15] Z. Wu, Z. Qin, J. Hou, H. Zhao, Q. Li, B. He, L. Fan, “Vertical Federated Learning in Practice: The Good, the Bad, and the Ugly,” 2025.
- [16] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, Q. Yang, “Vertical Federated Learning: Concepts, Advances and Challenges,” arXiv:2211.12814, v4 (2023)