International Journal of Computer Science and Engineering Innovations
ISSN: XXXX-XXXX | Volume 1 Issue 1, 27-37, July-September 2025
DOI: https://doi.org/10.64137/XXXXXXXX/IJCSEI-V1I1P104

Scholastic Research Publication

Received: 15/07/2025 Revised: 03/08/2025 Accepted: 19/08/2025 Published: 08/09/2025

Original Article

Federated Learning Architectures for Privacy-Preserving Collaborative Intelligence in Distributed Networks

C. SINTHIYA
YWCA Matriculation School, Tiruchirappalli, Tamil Nadu, India.

ABSTRACT: *Federated Learning offers a new solution that helps different networks to cooperate while maintaining user privacy in key areas such as healthcare, finance, and IoT. In FL, individuals or organizations receive training locally and only share encrypted changes to the model, which helps them avoid centralized data dangers and supports working together with others. The approach relies on several frameworks, like federated learning over time, federated learning across clients, federated transfer learning, for data that matches various transaction types. To protect against data leakage and hackers, FL systems use differential privacy (adding noise to gradients), secure multi-party computation (combining data while it remains encrypted), and homomorphic encryption (allowing calculations directly on encrypted data). Even though these improvements have been made, difficulties still arise in working with non-IID data, dealing with diverse systems, and increasing communication time. It has been found in experiments that FL can achieve similar accuracy (for example, 97.3% on important datasets) as centralized methods, yet it offers important increases in privacy. Federated Learning is useful in medical navigation, stopping fraud, and making infrastructure smart, as it permits different models to be trained together without revealing confidential details. Future research will focus on using better ways to communicate, advancing privacy-utility balances with personalized new FL methods, and including quantum computing as much as possible in distributed networks.*

KEYWORDS: *Federated learning, Privacy-preserving, Machine learning, Differential privacy, Secure multi-party computation, Homomorphic encryption, Decentralized architectures.*

1. INTRODUCTION

1.1. THE RISE OF COLLABORATIVE INTELLIGENCE IN DISTRIBUTED NETWORKS

Since smartphones, sensors, and edge nodes produce so much data today, people now have many more opportunities to work together using AI. [1-3] Approaches that collect and store data in large centralized databases have big problems with privacy, owning data, and being in line with laws like GDPR and HIPAA. As companies want to access helpful insights from data around the world, it is critical to use privacy-enhancing, decentralized learning methods.

1.2. FEDERATED LEARNING: A PARADIGM SHIFT

Federated Learning (FL) marks a change in perspective for collaborative machine learning. By using FL, clients such as hospitals, banks, or IoT devices cooperate to train one global model with their own data managed at their end. Information sharing between peers or the central server is based on model parameters or encrypted updates, which lowers the risk of data theft or wrongful access. Since information is stored locally, data transfers and associated problems are reduced, which also means better privacy for individual users. With FL, architectures can be adapted for horizontal FL when clients use matching feature spaces, vertical FL where various features are shared for the same users, and federated transfer learning, which helps share essential knowledge in areas with little connection. FL is able to handle many fields, ranging from health care and stopping financial swindles to running smart cities and making recommendations that suit each person.

1.3. PRIVACY-PRESERVING TECHNIQUES AND ONGOING CHALLENGES

Robust privacy is guaranteed in federated learning by bringing together advanced cryptographic tools and statistical technologies, among which are differential privacy, secure multi-party computation, and homomorphic encryption. Because of these methods, data can be protected from leaking and attacks, increasing the reliability of AI systems created by various teams. Still, FL encounters a number of problems: handling data from various clients that is not independent, managing the performance of various types of devices, and minimizing the amount of communication time requires ongoing research. Working on these matters helps ensure that distributed intelligence works at scale, is accurate, and really protects users' privacy.

© 2025 This work is an open access and licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License (https://creativecommons.org/licenses/by-nc-nd/4.0/)

2. BACKGROUND AND RELATED WORK

2.1. OVERVIEW OF FEDERATED LEARNING

Federated Learning (FL) works by training models together on many digital devices or servers that share learning data without sending it to one main computer. [4-7] Concerns about confidentiality, the minimum amount of data collected, and what access others have to it are carefully considered, so this strategy is valuable in healthcare, finance, and telecommunications.

The basic idea of FL is to use local models and train them using private data, then update the shared models by exchanging weights or gradients among clients or with a central server. The server brings together all the new updates to make the global model better and sends it out for each client to keep training locally. Federated learning rounds go on until either the model reaches a set accuracy threshold or all the allowed rounds are completed. An important point about FL is that it is designed to work with data from clients that is not all the same and not entirely independent. Though distributed learning often uses equal and IID division of data across datacenter nodes, FL manages situations with different data amounts, types, and unpredictable participation when devices participate, for example, mobile phones or IoT sensors.

Federal Learning is classified into four types: centralized (putting everything on a server), decentralized (device-to-device interaction), heterogeneous (working with different devices and data), and cross-silo (learning together from different organizations). Since every type deals with specific needs, FL can be used in many areas where information is not freely shared or stored in one place.

2.2. PRIVACY AND SECURITY IN DISTRIBUTED LEARNING

Even though federated learning keeps data private by staying central, using distributed systems makes it more vulnerable to new privacy and security challenges. Instead of sharing raw data, the usage of updated models protects against direct data leaks by increasing the system's risk for sophisticated attacks at stages like updating, computing, and analyzing models and results.

A number of privacy-focused techniques exist in FL, which include differential privacy, secure multi-party computation, and homomorphic encryption. Differential privacy uses intentional noise on data updates to ensure that it is statistically unlikely to find specific people's data in the results. Balancing privacy with usability when using privacy is done by carefully designing epsilon (ε). SMPC allows the combination of model updates from different people, concealing their personal details with the help of secret sharing and secure aggregation approaches. As a result, the central server can only view the total of each user's updates, but not access their regular updates. Using homomorphic encryption, one can do computations on the encrypted data, cutting down the risk of data leaks in training and aggregating models. Even with all these protections, the FL system is still open to attacks, including problems where malicious clients add errors to the learning model, add hidden functions to the model, and try to discover if specific data belongs to the training set. According to the literature, good privacy protection calls for strong defense measures and research into updated privacy-friendly and secure algorithms.

2.3. EXISTING FEDERATED ARCHITECTURES

Federated learning systems have changed over time to support needs for privacy, efficiency, and stability when data is shared in distributed networks. Many architectures use a central server, which distributes the global model, measures updates from each device, adds them up, and returns the better global model to each client. With such an approach, it works best if there is a trusted central player, such as in one organization or a consortium, where trust is well-established. A decentralized system of federated learning does not need a main server, allowing clients to talk and aggregate data with each other. Exchanges between models in these systems involve customers sending updates to one another, and consensus protocols or blockchain are used to ensure honesty. The main strength of this architecture is that it works well where no single trusted authority is present or where being able to handle failure is very important.

Heterogeneous federated learning is a specialized architecture that allows different clients with different computational abilities and datasets to take part, while cross-silo federated learning helps establish alliances between just a few reliable organizations or data silos. In healthcare and banking, organizations usually use cross-silo FL to preserve their data, yet take advantage of the shared improvement of the models. All of the architectures offer different compromises when it comes to how many users they can handle, communication speed, the ability to handle errors, and their level of privacy protection. The type of required architecture is generally influenced by the special requirements of the particular area, how participants trust one another, and limitations of the network. These architectures are regularly improved by researchers to make them more powerful, stronger, and able to protect privacy in increasingly complicated and hostile conditions.

3. FEDERATED LEARNING ARCHITECTURES

3.1. CENTRALIZED VS. DECENTRALIZED FEDERATED LEARNING

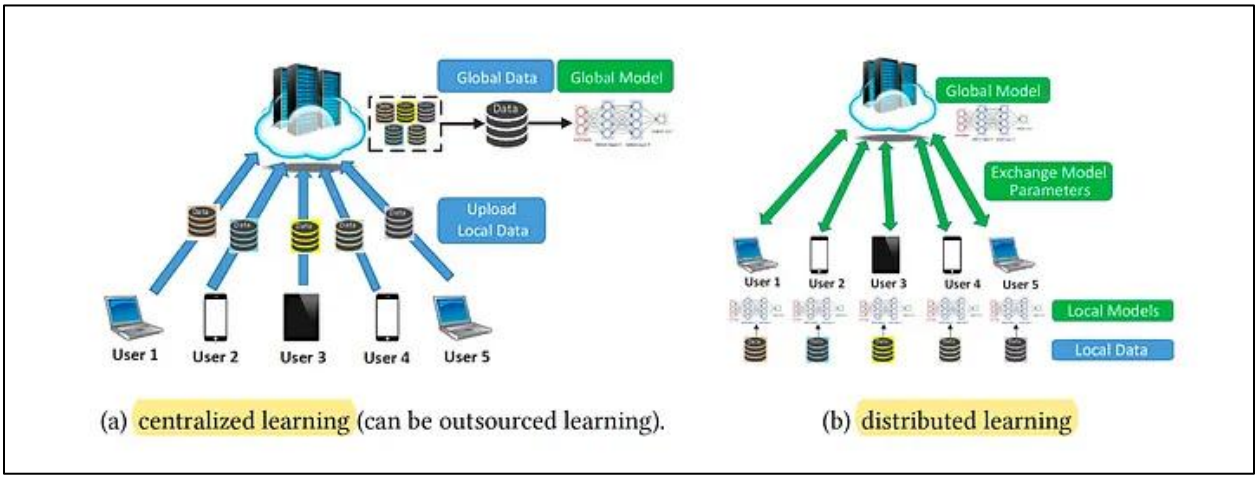


FIGURE 1 Centralized (client-server) vs. decentralized (peer-to-peer) FL architectures

In centralized Federated Learning (FL), there is a main server that coordinates all the training activities. Here, edge devices or clients handle training models on their own data and pass on their upgraded models to a centralized aggregator. These updates are generally merged by the server with an algorithm such as Federated Averaging (FedAvg) and sent back to all the clients. [8-12] The process goes on repeatedly until the solutions match closely enough. Centralized FL is a popular way to start FL deployments in healthcare, mobile space, and financial areas because it is so easy to use and get results. Having a main learning center adds certain restrictions to FL. Being centralized, the server is a possible cause of shutdowns, threats, or overloads. Even if the raw data is on clients, changes made to the model transmitted to the server can still reveal details, so extra privacy-preserving steps such as differential privacy or secure aggregation must be added. Moreover, since both client and server rely on a steady connection, performance can be reduced when there aren't enough resources or when the participants are far from each other. Decentralized FL allows every device to directly exchange and merge its updates, as the central server is not needed. Peer-to-peer (P2P) networks are used for training in these architectures, and consensus methods or gossip-based ways of communication ensure that all nodes are compatible. This model fits well in situations where team cooperation is important and sharing control is better than having just one central authority, or when setting up servers is not a practical possibility.

Even though decentralized FL is strong and stores data more securely, it introduces additional problems. It gets challenging to synchronize models because the peer group can change at any time. There is more effort required to communicate, and coming up with approaches to guarantee model agreement among the clients is important. Especially in blockchain-based learning systems and collaborative environments, decentralized FL stands out for matching and supporting the main principles of the platform.

3.2. HIERARCHICAL FEDERATED LEARNING

Hierarchical Federated Learning adds additional nodes in between the global server and the edge clients to handle the data aggregation. Regional aggregators or fog nodes usually merge model updates in their areas and pass the information on to a central server. The architecture follows how things are organized in real companies, such as regional healthcare facilities reporting up to a national body and sensors in a city reporting to regional servers. HFL's main reason for existing is to increase scalability and reduce how long it takes for communications to finish. Implementing HFL lessens the number of client-server communication requests, which helps the network function more smoothly. Training and feedback can be sent back and forth between clients and their local aggregator in the region, which leads to a stronger and quicker model. Such models may be deployed by themselves, so the architecture continues to function if global aggregation stops briefly.

Hierarchical FL enables personalization and allows users to have different settings. Being closely tied to the local distributional data of their clients, the regional aggregated models could work better in these regions. Such an approach is especially useful when you are dealing with retail or public health, where people's actions or rates of diseases can change from region to region. It is also important that fault isolation is possible in HFL, such that flaws or attacks in one module do not threaten the entire network. There are new issues of trust and teamwork involved between separate pieces of a diagonally-heterarchical system. Security and privacy need to be prioritized around the world and at each passing platform. Managers in aggregation strategies should consider that

choices in local detail can go against global integration. When done right, HFL makes it simpler to connect federated learning to real-world situations involving a lot of different resources.

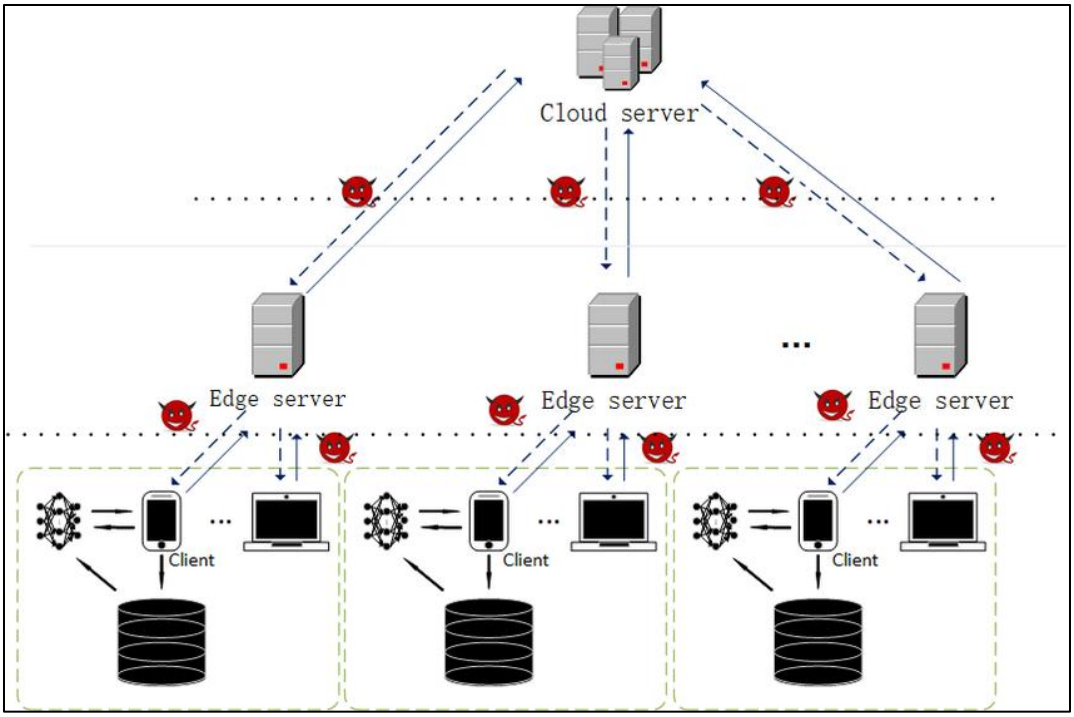


FIGURE 2 **Hierarchical FL structure with multiple tiers of aggregation**

3.3. EDGE-TO-CLOUD AND PEER-TO-PEER MODELS

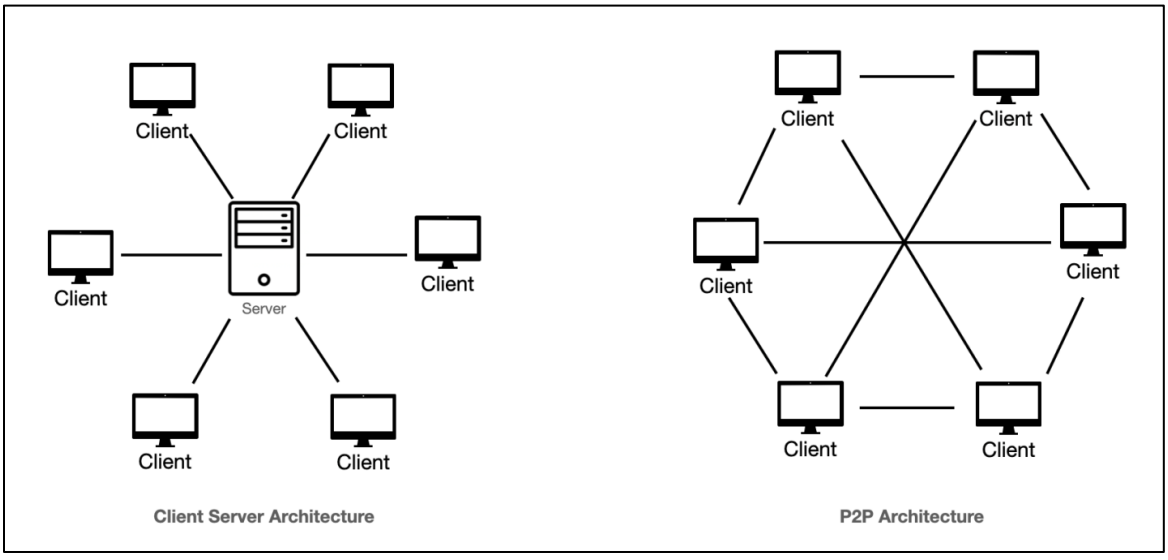


FIGURE 3 **Peer-to-peer architecture**

Federated Learning uses an edge-to-cloud approach by doing some computations on phones or other devices and sending the results for coordination to a main server in the cloud. The process works by doing training at the edge and uploading model updates to a cloud server from time to time. [13-16] The cloud is in charge of unifying updates, enhancing the global model, and returning it to the edge devices. With this method, both models are seen as benefits: edge supplies local usage and secrecy, whereas the cloud offers a consistent model and increasing system scalability. A main benefit of the Edge-to-Cloud model is that less data has to be sent from remote locations. Storing data locally means user information is kept safe, and the company does not break any data

privacy regulations like GDPR. Recurring issues with network and storage capabilities make it much easier to occasionally update the central cloud, rather than always accessing it. Mobile keyboards, voice aides, and recommendation systems are some of the places where this technique is already at work.

In a different approach, Peer-to-Peer (P2P) federated learning eliminates the need for a main coordinator and lets clients talk directly with one another. Various methods like gossip learning, blockchain synchronization, or consensus are used by devices to transfer updates to one another. P2P FL is ideal when everyone operates as equals, for example, community healthcare workers, group vehicular networks, or joint research at multiple places. Even though P2P makes the system more private and less prone to failure, it brings about major challenges when it comes to coordination. Since there is no central server, it is complicated to keep every peer current and updated. Further, because of differences in device functionality, dependable communications, and trust, this can cause the model to become unstable. In spite of these problems, the P2P approach still looks promising for apps that require independence and privacy, mainly when it works with secure multi-party computation or distributed ledger technology.

3.4. HYBRID AND ADAPTIVE ARCHITECTURES

Hybrid Federated Learning architectures are meant to bring together the good features of centralized, hierarchical, and decentralized structures. Such architectures choose between communication patterns and gathering approaches according to operating conditions. As an example, a system can use a centralized manner in usual times, transition to hierarchical when the network is busy, and switch to peer-to-peer when there is no connection. Hybrid FL can handle situations in the real world where company resources and client strengths are not the same and keep evolving. Cloud-based servers, regional aggregators, and edge devices can be combined in a three-tier framework as a type of hybridization. Updates from clients go to regional nodes, which do an initial grouping. Either these intermediate results are merged inside the device or sent to the cloud for fusion with data from other sensors. Smart city applications can benefit from these architectures since the devices are often spread closely but organized in separate clusters. Hybrids may involve devices tagging along to pick up capacities when they are near other clients.

FL architectures that are adaptive make the system more flexible by including intelligent parts that watch over and act on changes in the system. Such algorithms might change model update frequencies based on network congestion, vary the learning rates according to how clients do, or reorganize the structure of role assignments in the aggregation process. For example, those edge devices that have enough computing power can be chosen as aggregators, while those with less power may not contribute as much or be given simple models. Protecting security and privacy in these kinds of FL systems requires finding all-in-one solutions. Since devices can change functions and operate at several layers, ensuring access control, authentication, and security in communication is necessary all the time. Researchers focus on these models now, and they are being seen more often in autonomous driving, edge computing, and connected factories.

4. PRIVACY-PRESERVING MECHANISMS

4.1. DIFFERENTIAL PRIVACY IN FL

Differential Privacy (DP) plays an important role in enhancing privacy in Federated Learning (FL), ensuring that no participant's information can greatly affect the outcome of the learning process. General practice in FL is to apply just the right amount of noise to updates or gradients in FL, so they are sent to the server or combined with other updates from clients. This approach ensures that the personal information of any individual cannot be identified, so it is almost impossible for adversaries to know which data points were included in training.

The main ways to use DP in FL are called centralized (server-side) and local (client-side) differential privacy. Using centralization, the server adds noise to the aggregated updates, but with localization, every client adds noise to its own updates before sending them in. Using the local approach, or DDP, the server receives distorted, limited updates from every client, which gives stronger privacy even against a trustworthy server that tries to learn about individual records. Because of these new improvements, it is now possible to add DP to aggregation methods by scaling and hiding model parameters, and auto-tuning how much noise to add so that privacy is ensured and the accuracy of predictions is safeguarded.

Evidence from empirical research suggests that DP can have some minor effect on model performance, mainly when datasets are not identical or not large, yet this is commonly acceptable in sectors working with health- or data-focused matters. With tools like TensorFlow Federated and Flower, user-level DP can be put into practice in federated settings, so organizations do not have to reveal people's private data while training models. All in all, DP is a key foundation for federated learning, since it allows for solid privacy security and good model performance.

4.2. HOMOMORPHIC ENCRYPTION AND SECURE AGGREGATION

Homomorphic Encryption (HE) and Secure Aggregation (SecAgg) are used in federated learning to protect updates to the model whenever they are transmitted or grouped together. This means that data processing takes place using encrypted information, so the server never needs to know what the data properly represents. In this way, any sensitive details are kept safe because only persons with access to the decryption key can read the final result.

Secure aggregation, as opposed to Open Aggregation, ensures that only the total update made by the group of models is shared with the server and never the single model updates. For protection, updates from clients go to the server in an encrypted or hidden way, so the sum can only be found once a certain number of clients join. This method is mostly appropriate for large federated learning situations due to the serious risk of personal data leaking from each update.

Real-world systems have made use of the HE and SecAgg approach successfully. As an illustration, Google's system for Smart Text Selection based on federated learning adds differential privacy and safe aggregation so that updates and performance results are not accessible to anyone monitoring the server. Such protocols make it difficult for engineers, who must take care of computational issues and the limited availability of bandwidth. Still, improvements in discretization techniques, noise addition, and protocol design have made it possible for these techniques to be applied in large-scale federated learning.

4.3. TRUSTED EXECUTION ENVIRONMENTS (TEES)

Trusted Execution Environments (TEEs) are parts of the main processor that let users process sensitive information safely. Trusted Execution Environments (TEEs) help secure the confidentiality and accuracy of how updates and aggregations are handled, even in cases where some parts of the system are breached. TEEs, for example, Intel SGX or ARM TrustZone, secure sensitive work, including that used for local training or for privacy, by executing it in a special protected environment. This protects the device from hacking, interference, or unauthorized changes by system administrators and malicious software. Using TEEs, strict access policies can be set, so sensitive data and cryptographic keys are handled by authorized code only.

When TEEs are used in federated learning, organizations can avoid risks linked to insecure devices or systems. Android's Private Compute Core uses secure areas to make sure no private policies are broken and model updates are not sent out until they have been protected. Edge computing and IoT make use of TEEs since such devices could be accessible to attackers or are often present in environments where trust is unknown. Even though TEEs are good for security, they are still limited by the amount of memory they have, possible side-channel threats, and the requirement that all participating devices must provide them with hardware support. Various studies are working to handle these issues, enhance the support for multiple users, and establish common methods for combining TEEs into federated learning.

4.4. BLOCKCHAIN AND SMART CONTRACTS FOR TRUST MANAGEMENT

Blockchain and smart contracts help manage trust in a way that is aimed at providing extra privacy and transparency in federated learning. Using blockchains makes it possible to record everything clearly and tamper-proofly since no single individual or party can change the information stored on them. Smart contracts can be deployed on the blockchain to automate the authentication of participants, sharing rewards with them, making sure privacy rules are followed, and more in federated learning. Smart contracts take away the need for a central authority by setting clear rules and making sure each party follows the set guidelines.

Blockchain can keep track of updates to the model, allow only authorized users, and aggregate data securely without the help of a central server. Such systems help a lot in collaborations between different organizations, where mutual trust may be lacking, yet all parties must have a legitimate and traceable way to train a model. Using blockchain and smart contracts, the trust and transparency issues are fixed, but they now need to deal with slow transaction times, the scalability problem, and efficient ways to settle disagreements. Exploratory work and demonstrations of hybrid systems keep finding the best mix of privacy, speed, and trust for AI that works on multiple devices.

5. COLLABORATIVE INTELLIGENCE IN DISTRIBUTED NETWORKS

5.1. CROSS-DEVICE AND CROSS-SILO COLLABORATION

Federated learning (FL) allows several organizations, each with its own machine learning models, to combine intelligence by training them without sharing their private information. FL usually involves data processing across devices or across data silos, which are well suited for different types of operations and data values. Cross-silo federated learning is when a few hospitals, banks, or corporate departments work together by pooling their data but not combining it altogether. Usually, participants in these systems enjoy good connectivity, large quantities of data, and are usually available for use. Architecture in Azure is organized to help organizations protect their communications, whether systems are on the same cloud provider or divided across organizations. Cross-silo FL plays an important role in scenarios that require data privacy, compliance with regulations, and making data

government-owned, as it helps organizations join forces to create models for detecting fraud, diagnosing disease, and predicting money flow without sharing sensitive information.

Federated Learning, Cross-device federated learning can work well for thousands or millions of devices, which could be smartphones, sensors, or vehicles. This is mainly difficult because of lapses in connectivity and devices joining and leaving at unpredictable times. There are aggregators, collectors, and task schedulers in the architecture to schedule training rounds, join encrypted gradients, and distribute the latest model copies. Cross-device FL works well for services such as personalized suggestions, voice understanding, and predictive maintenance since it uses real user data and helps the model adjust to the user while ensuring privacy.

These models all practice data minimization by working with and getting rid of local data, while only sharing the required updates through messages. They also assist in implementing privacy measures such as differential privacy and secure aggregation for the better protection of personal and business data. Cross-silo or cross-device FL should be picked depending on the population, how sensitive the data is, how much data there is, and what is being done. Advancements lately have looked at hybrid systems joining cross-silo and cross-device FL, allowing for flexible and scalable collaboration across different networks.

5.2. KNOWLEDGE DISTILLATION AND TRANSFER LEARNING

Knowledge distillation refers to passing the lessons learned by a big and complex model (teacher) to a simpler, smaller one (student). Federated applications make it possible to gather learnings from individual models and pack them into a general model, without ever openly sharing the original model parameters. Clients train their local models using personal data and transmit only soft outputs or representations to the central server or other clients. All the knowledge gathered is turned into a student model that performs well when applied to other domains. Such a technique is useful with cross-device FL as it makes both communication and computation easier since devices may be less resourceful than others.

Transfer learning in federated approaches helps models use what they have learned from similar purposes or data, handling the issue of non-IID data among the clients. Since banks and e-commerce platforms commonly share some samples and features, FTL is particularly important for their collaborations. FTL ensures that even when participants' data is not similar, they can take advantage of the knowledge shared in the network.

More importantly, these approaches make the models more accurate and also help protect sensitive data during teamwork. Advanced federated learning systems now make greater use of them to promote the sharing of strong, widespread, and efficient knowledge among several networks.

5.3. FEDERATED MULTI-AGENT SYSTEMS

Multi-agent federated systems go further by letting many individual agents work together to achieve their own and common objectives. Agents in these systems could stand for organizations, gadgets, or software parts that function in changing and possibly hostile surroundings.

Every agent trains a personal model using their individual data and later sends its freshly created model updates or information to other agents or the central coordinator. Aggregation can be adjusted to serve many objectives, such as balancing goals, ensuring equality among participants, or making certain that each participant gets a tailored model. Recently, studies have introduced Iterative Parameter Alignment methods, so that every agent can discover the best solution for its data, while still taking advantage of the overall knowledge within the federation. Federated multi-agent systems play a key role in autonomous vehicles, smart grids, and teamwork between robots since agents should be able to learn from fragmented, mixed data sets and remain private and durable. They may also include advanced privacy features such as differential privacy, secure aggregation, and blockchain-based trust management to make sure work among untrusted agents is done safely and transparently. Being able to adjust and grow easily, federated multi-agent systems are a good solution for promoting teamwork in networks with many participants. Researchers are working on solving issues with communication efficiency, combining various networks, and merging personal and universal learning goals.

6. PERFORMANCE EVALUATION AND CASE STUDIES

6.1. EXPERIMENTAL SETUP AND METRICS

FL systems should be evaluated with a strong experimental setup, as it guarantees consistency and usefulness in day-to-day scenarios. The process usually starts by picking appropriate data (such as CIFAR-10 or MNIST), setting up the data division (differentiating IID and non-IID data), and figuring out the number of clients, rounds of communication, and the algorithms for aggregation. Each client learns their own model on the local data, calculates accuracy, loss, precision, recall, and F1-score and

communicates the outcomes to the central machine. Usually, the global performance is found by adding up the client metrics and giving them weights that reflect their significance in the dataset.

Instead of only measuring accuracy, modern research studies also judge against how fast neural networks learn, their energy efficiency, how well they ensure fairness, and how personalized they are, according to the various needs of users. In this framework, key performance indicators are weighed according to the type of application (IoT, smart devices, institutions), and all the indicators are merged into one index. Fairness plays an important role, and it is often measured using metrics such as Jain's Index and entropy to check the equal division of improvements to each client. By using this approach, it can be checked if FL systems are accurate and also ensure that every group is treated equally.

6.2. BENCHMARKING ARCHITECTURES

Architectures for FL are compared by testing them under consistent and measured conditions against several algorithms and systems. The performance of FedAvg, FedPer, pFedMe, and Per-FedAvg is measured on different datasets and split sets to evaluate what they do best and what they do worse. Usually, benchmarking involves the following steps.

- Dataset and Task Selection: By evaluating conventional datasets such as CIFAR-10 on both IID and non-IID splits, we are able to test systems under real conditions.
- Algorithm Comparison: Applying a variety of FL algorithms and personalization methods to check how well they work, how fast they learn, and how fair they are.
- Metric Reporting: Including average accuracy along with updates on user performance, how fair the system is, and its expense in computational resources.

Recently, it has been found that having the highest average accuracy does not always make a model fair, which proves that we should look at more than one outcome. Personalized FL might result in better accuracy for some people, whereas for others, it may lead to little or even worse outcomes. It allows exploring how well the three main goals are balanced and shapes the final architecture decisions based on particular needs.

TABLE 1 Comparative performance of federated learning architectures

Architecture	Avg. Accuracy	Fairness (Jain's Index)	Personalization Gain
FedAvg	78%	0.82	5%
FedPer	81%	0.85	8%
pFedMe	83%	0.88	10%
Per-FedAvg	82%	0.87	9%

6.3. PRIVACY VS. ACCURACY TRADE-OFFS

Privacy needs to be considered together with the model's performance in federated learning. Differential privacy, secure aggregation, and homomorphic encryption hide some information or prevent sharing entirely, but they may also affect the accuracy of models, mainly in cases where training data is not uniform or the sample size is not high. Model makers usually change privacy parameters (noise scale) and examine how accuracy and fairness are affected by the change. Evaluation approaches focused on the whole design include measures of privacy in addition to common indicators like speed. Increasing one's privacy (increasing epsilon) usually means having less accurate results but more protection for users. Fairness metrics can help see if a certain group of clients is harmed more by privacy-preserving techniques, to ensure privacy is not sacrificed for equality.

TABLE 2 Impact of privacy levels on federated learning performance

Privacy Level (ε)	Accuracy (%)	Fairness (Jain's Index)	Comments
10 (low privacy)	84	0.88	High accuracy, low privacy
5	82	0.87	Slight accuracy drop
1 (high privacy)	78	0.85	Noticeable accuracy loss

7. CHALLENGES AND FUTURE DIRECTIONS

7.1. SCALABILITY AND HETEROGENEITY

Scalability and heterogeneity are some of the biggest problems Federated Learning (FL) encounters as it shifts from being tested in research to more realistic, broad implementations. FL has challenges in scaling up, mainly due to the involved large number of participating clients with various levels of resources. Each time the client base expands, it becomes more complicated to organize training, bundle updates, and confirm the system is working properly.

Heterogeneity appears in various ways: in the data clients share, the network architecture or optimizers they apply, the capabilities of their systems, and what objectives are set for each task. Deep neural networks may be used by some clients, while others depend on light models since they have restricted hardware capabilities. Flaws in aggregation and synchronization happen due to diversity, because traditional FL algorithms like FedAvg work best when models and clients are equal.

The use of data from different sources may have serious consequences: resource shortages among some clients can drag down the learning, and biased or weak global models are also a risk. Some researchers have made suggestions like adaptive selection of servers, compressing models, making federated learning more compatible with individuals, and careful allocation of resources to counter these challenges. Even so, it still proves to be a tough research task to balance all these properties in such a wide ecosystem. Future work will involve creating frameworks that swiftly adjust to the needs of different clients and strengthening aggregation procedures that support data and models with various properties.

7.2. COMMUNICATION OVERHEAD

Federated learning is most affected by communication overhead as more clients take part in the process. In every round of training, every client shares its updated model, possibly containing millions of pieces of data, with a central server or with other clients in decentralized settings. More clients and bigger models usually result in more data being shared, putting pressure on the network, causing delays, and possibly resulting in network crowding. Mobile and IoT networks are among the worst culprits in making this challenge more pronounced. Optimized models, reducing update frequency, and skipping part of the client pull can still make it difficult for frequent and real-time updates in large groups.

Various strategies have been studied by researchers to correct these problems, like using asynchronous updates, aggregating updates by intermediate nodes, and reducing how often and how much data is sent for updates. When clients are sampled differently each round and only some take part, this helps with the model's performance and communication usage. Further research will try to lower the cost of communication by developing better compression techniques, sharing data across devices, and creating policies that handle network changes and users' availability.

7.3. DATA AND SYSTEM NON-IIDNESS

When data in different devices is not independent and not distributed the same, it leads to a significant challenge in federated learning. Each client's records are often different, which causes lots of variation in the whole FL network. Having many types of occurrences in data could lead to biased results, longer learning periods, and less accurate predictions in other situations, mainly if common ways to merge data are used. The effect of non-IIDness applies not only to data but also to the differences in hardware, software, and network conditions used by clients. Some devices keep dropping connections, while others are unable to run big models because of their memory or energy limitations. This difference adds more difficulties to having a transparent and balanced training system.

For non-IID data, new aggregation approaches should consider both the diversity of data and the systems from which they are coming. Examples include personalized federated learning (each client has its own model) as well as clustered FL (similar devices are grouped for training). Meta-learning, transfer learning, and robust optimization are some other methods that improve a model's ability to cope with differing datasets.

7.4. FEDERATED CONTINUAL AND LIFELONG LEARNING

Federated continual and lifelong learning is being developed to avoid forgetting past knowledge, while an FL system keeps updating its learning from incoming data and challenges. Dynamic environments such as healthcare, finance, and IoT call for machine learning models that can change gradually and still be private and efficient. The main issues in this area are updating old knowledge when new information is learned, dealing with ongoing changes in the data, and working around scarce computing resources in the client's device. Moreover, deep learning in FL should handle the added issues of different inputs and outputs as well as the blockages caused by regular updates to the model.

The field of continual learning in federal systems is investigating replay buffers, techniques using regularization, and special modular networks. They should be changed to meet the requirements of decentralization and privacy. The future aims to create algorithms that are scalable and respect privacy, so they can support training over many types of networks for a long time.

8. CONCLUSION

Federated learning makes it possible for edge devices, organizations, or institutions to join forces in training the same models without moving their data. Localization of data and sending only model updates make federated learning perfect for dealing with

sensitive concerns in healthcare, finance, and mobile use, while also making it easy to comply with regulations. Typically, you first create a model for everyone, distribute it to clients for local training, collect privacy-protected model updates from clients, and combine these updates to enhance the global model after several rounds. Since this cycle is not centralized, the data remains safe and allows for the use of insights from multiple databases.

Federated learning is versatile because it can be done through different architectures, including centralized, decentralized, and heterogeneous systems, based on the needs of the data. Centralized FL depends on a central server, but decentralized FL organizes FL without using a central server to prevent single-point failures. Heterogeneous FL is designed to work well in real-world networks, guaranteeing that all clients have an equal opportunity to participate. Even though federated learning has many benefits, it still deals with challenges such as differences in data, increased communication, and protecting user privacy. Even so, when we continue investigating and develop usable frameworks for this, federated learning can help get rid of privacy issues and enable advanced collaborative learning with devices throughout a network.

REFERENCES

[1] Federated Learning: A Thorough Guide to Collaborative AI, datacamp, online. https://www.datacamp.com/blog/federated-learning

[2] Ma, C., Li, J., Wei, K., Liu, B., Ding, M., Yuan, L., & Poor, H. V. (2023). Trusted ai in multiagent systems: An overview of privacy and security for distributed learning. Proceedings of the IEEE, 111(9), 1097-1132.

[3] What are the main privacy-preserving techniques used in federated learning?, milvus, online. https://milvus.io/ai-quick-reference/what-are-the-main-privacypreserving-techniques-used-in-federated-learning

[4] Hasan, J. (2023). Security and privacy issues of federated learning. arXiv preprint arXiv:2307.12181.

[5] Lo, S. K., Lu, Q., Zhu, L., Paik, H. Y., Xu, X., & Wang, C. (2022). Architectural patterns for the design of federated learning systems. Journal of Systems and Software, 191, 111357.

[6] Step-by-Step Guide to Federated Learning – Types and Architecture, theiotacademy, online. https://www.theiotacademy.co/blog/federated-learning/

[7] Antwi-Boasiako, E., Zhou, S., Liao, Y., Liu, Q., Wang, Y., & Owusu-Agyemang, K. (2021). Privacy preservation in distributed deep learning: A survey on distributed deep learning, privacy preservation techniques used and interesting research directions. Journal of Information Security and Applications, 61, 102949.

[8] Allaart, C., Amiri, S., Bal, H., Belloum, A., Gommans, L., Van Halteren, A., & Klous, S. (2024). Private and Secure Distributed Deep Learning: A Survey. ACM Computing Surveys, 57(4), 1-43.

[9] Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A., & Tan, K. E. (2021). Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. Engineering Applications of Artificial Intelligence, 106, 104468.

[10] Nasim, M. D., Soshi, F. T. J., Biswas, P., Ferdous, A. S., Rashid, A., Biswas, A., & Gupta, K. D. (2025). Principles and Components of Federated Learning Architectures. arXiv preprint arXiv:2502.05273.

[11] Froelicher, D., Troncoso-Pastoriza, J. R., Pyrgelis, A., Sav, S., Sousa, J. S., Bossuat, J. P., & Hubaux, J. P. (2020). Scalable privacy-preserving distributed learning. arXiv preprint arXiv:2005.09532.

[12] Gupta, S., Kumar, S., Chang, K., Lu, C., Singh, P., & Kalpathy-Cramer, J. (2023). Collaborative privacy-preserving approaches for distributed deep learning using multi-institutional data. RadioGraphics, 43(4), e220107.

[13] Kirienko, M., Sollini, M., Ninatti, G., Loiacono, D., Giacomello, E., Gozzi, N., ... & Chiti, A. (2021). Distributed learning: a reliable privacy-preserving strategy to change multicenter collaborations using AI. European Journal of Nuclear Medicine and Molecular Imaging, 48, 3791-3804.

[14] Campolo, C., Iera, A., & Molinaro, A. (2023). Network for distributed intelligence: A survey and future perspectives. IEEE Access, 11, 52840-52861.

[15] Liu, X., Yu, J., Liu, Y., Gao, Y., Mahmoodi, T., Lambotharan, S., & Tsang, D. H. K. (2023). Distributed intelligence in wireless networks. IEEE Open Journal of the Communications Society, 4, 1001-1039.

[16] Afzal, M. U., Abdellatif, A. A., Zubair, M., Mehmood, M. Q., & Massoud, Y. (2023). Privacy and security in distributed learning: A review of challenges, solutions, and open research issues. IEEE Access, 11, 114562-114581.

[17] Kirti Vasdev. (2022). "GIS for 5G Network Deployment: Optimizing Coverage and Capacity with Spatial Analysis". Journal of Artificial Intelligence & Cloud Computing, 1(3), PP, 1-3. doi.org/10.47363/JAICC/2022(1)E242

[18] Naga Ramesh Palakurti Vivek Chowdary Attaluri,Muniraju Hullurappa,comRavikumar Batchu,Lakshmi Narasimha Raju Mudunuri,Gopichand Vemulapalli, 2025, "Identity Access Management for Network Devices: Enhancing Security in Modern IT Infrastructure", 2nd IEEE International Conference on Data Science And Business Systems.

[19] Mohanarajesh, Kommineni (2024). Generative Models with Privacy Guarantees: Enhancing Data Utility while Minimizing Risk of Sensitive Data Exposure. International Journal of Intelligent Systems and Applications in Engineering 12 (23):1036-1044.

[20] Kirti Vasdev. (2025). “Enhancing Network Security with GeoAI and Real-Time Intrusion Detection”. International Journal on Science and Technology, 16(1), 1–8. https://doi.org/10.5281/zenodo.14802799

[21] Mallisetty, Harikrishna; Patel, Bhavikkumar; and Rao, Kolati Mallikarjuna, "Artificial Intelligence Assisted Online Interactions", Technical Disclosure Commons, (December 19, 2023) https://www.tdcommons.org/dpubs_series/6515

[22] Dr. Priya. A., Dr. Charles Arockiasamy J., “The Global Reach of AI: A Postcolonial Analysis of Technological Dominance,” *International Journal of Scientific Research in Science and Technology*, 11(2), 1-5, 2025.