Definition: What is Federated Learning?

Federated Learning (FL) is formally defined and described across the literature as:

- A machine learning setting where **many clients** (e.g., mobile devices or whole organizations) **collaboratively train a model** under the orchestration of a central server (e.g., service provider), while **keeping the training data decentralized** [4].
- A learning paradigm introduced in 2016 by McMahan et al. where the learning task is solved by a loose federation of participating devices (clients) coordinated by a central server [4].
- A mechanism that embodies the principles of **focused collection and data minimization** [4]. Client raw data is stored locally and is explicitly not exchanged or transferred [4]. Instead, **focused updates intended for immediate aggregation** are used to achieve the learning objective [4].
- A model that mitigates many of the **systemic privacy risks and costs** resulting from traditional, centralized Machine Learning (ML) and data science approaches [4].
- A **privacy-preserving decentralized approach**, which keeps raw data on devices and involves local ML training while eliminating data communication overhead [1]. A federation of the learned and shared models is then performed on a central server to aggregate and share the built knowledge among participants [1].

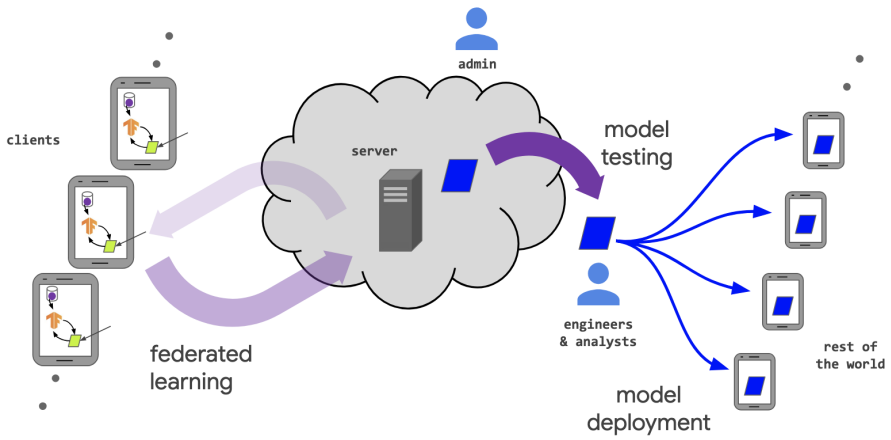

Figure 1: the various actors in a federated learning system [4]

Mechanism: How does Federated Learning work?

The standard FL workflow is orchestrated by a central server and is often based on the **Federated Averaging (FedAvg) algorithm** [4, 1].

Architecture and Workflow

The FL life cycle consists of several continuous communication rounds until the global model reaches the desired accuracy [1].

1. **Client Selection/Initiation:** The server generates or initializes a generic model [1]. In each round, a subset of clients is sampled [4]. Typical eligibility requirements include the device being **in charge, idle, and on an unmetered connection** (i.e., Wi-Fi) [1]. Advanced protocols like **FedCS** may require clients to report information about their resources (upload and update time) to a Mobile-Edge Computing (MEC) server, which then determines the subset able to complete the FL steps within a deadline [1].
2. **Broadcast and Download (Phase 1 in HI):** Selected clients download the current model parameters/weights and a training program (e.g., a TensorFlow graph) from the server [4, 1]. The global updates might propagate to local nodes at different time instants depending on communication channel specifics [3].
3. **Local Training (Phase 2 in HI):** Each device locally trains and optimizes the global model using its local data [1]. This involves running SGD [1].

4. **Upload:** The client sends its locally computed parameters/updates to the central server [1].
5. **Aggregation (Phase 3/4 in HI):** The server collects and aggregates the updates. This process is often a **weighted average** based on client dataset size [1, 3].
6. **Model Update (Phase 4 in HI):** The server uses the aggregated gradient to update the global model. This new shared model is distributed back to the clients, and the process repeats until convergence [1, 3].

In this star network communication topology, a subset of selected devices performs local training on their **non-identically-distributed user data** and sends these local updates to the server [2].

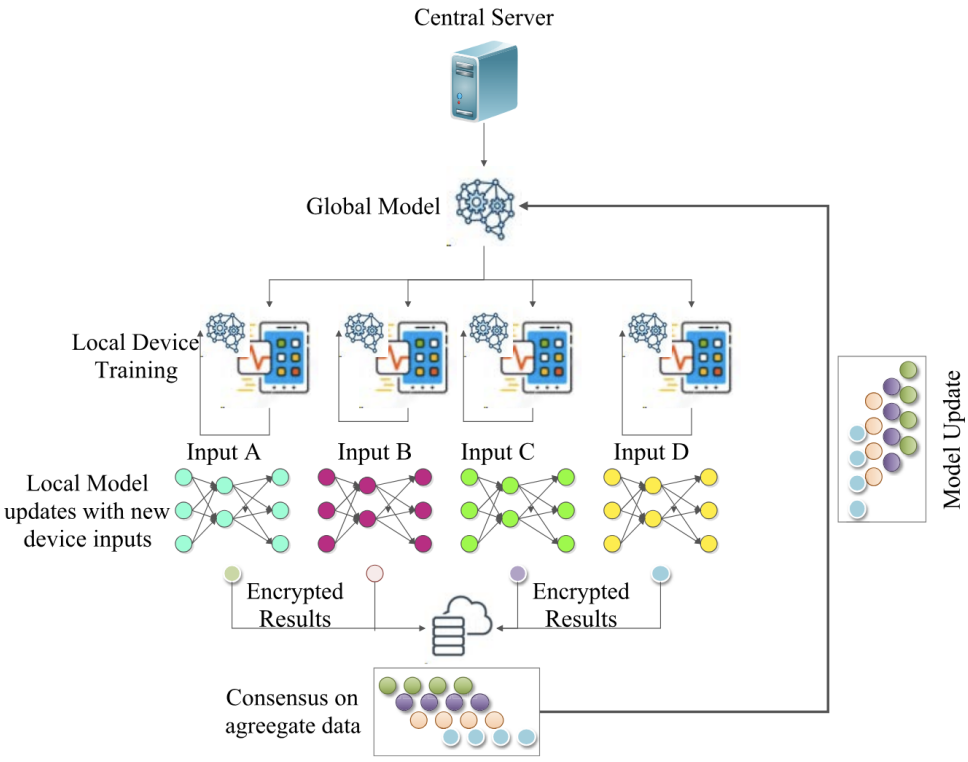

Figure 2: FL training process.

Alternative Learning Paradigms and Techniques

- **Fully Decentralized / Peer-to-Peer Distributed Learning:** This approach removes the need for a central server coordinating the overall computation, with devices communicating directly with neighbors [4].
- **FL Algorithms for Aggregation:**
  - **FedPer (Federated learning with personalization layer):** Clients send base layers (for representational learning) to the server but retain higher layers (decision specifics) locally [3].
  - **FedMA (Federated learning with matched averaging):** Constructs a global model by averaging values of hidden layers (e.g., convolutional layers) [3].
  - **FedDist (Federated Distance):** Computes the distance between neurons with similar features, suitable for sparse data [3].
  - **One-shot FL:** The massive number of communication rounds is substituted with only **one round** where each device trains its model until completion, and models are aggregated using ensemble learning techniques [1].
- **Data Partitioning:**
  - **Horizontal FL (HFL):** Local clients have the **same feature sets** but in **different data spaces** [3].
  - **Vertical FL (VFL):** Local clients have the **same sample space** but **different feature sets** [3].

– **Transfer Learning-based FL (TL-FL):** Extends VFL to include more local clients with different feature sets but the same sample space, extracting common features to a shared space, suitable for EHRs collected from heterogeneous sources [3].

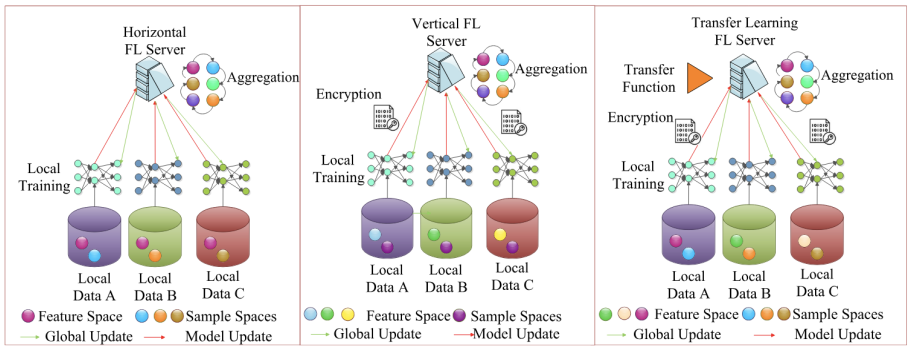

Figure 3: FL-HI aggregation conceptual overview: The left figure depicts the horizontal FL, centre figure depicts the vertical FL type, and the right figure indicates the transfer learning scheme in FL [3]

Applications

FL is deployed across massive consumer products (Cross-Device) and organizations with high confidentiality needs (Cross-Silo).

Consumer, Mobile, and Edge Devices (Cross-Device FL)

- **Mobile Keyboards:** FL was first tested on Google's virtual keyboard, **Gboard** [1]. It is used to improve **next-word prediction**, **query suggestion**, word completion, corrections [1], and **emoji prediction** [1]. FL is also adapted to learn **out-of-vocabulary (OOV) words** [1].
- **Mobile Operating Systems:** Used for features on **Pixel phones** and in **Android Messages** [4]. Apple uses cross-device FL in **iOS 13** for applications like the **QuickType keyboard** and the **vocal classifier for "Hey Siri"** [4].
- **General Mobile Apps:** Powers applications such as face detection and voice recognition [2]. Explored for **hotword detection** [4].
- **IoT Systems:** Used to limit vulnerabilities in large-scale IoT systems [1]. Use cases include **intrusion detection systems** based on anomaly detection [1] and **Mobile Edge Computing (MEC)** optimization [1].
- **Electric Vehicles (EVs):** Analyzing driver behavior metrics (using LSTM) to predict battery failure [1].

Organizational and Industrial (Cross-Silo FL)

- **Healthcare Informatics (HI):** FL is a strong fit for HI, balancing patient privacy with ML by keeping patient data on-premise [1]. Applications include:
  - Predicting mortality and hospital stay time using distributed **Electronic Health Records (EHRs)** [1].
  - Predicting hospitalizations for heart disease patients [1].
  - **Medical image prediction** (e.g., brain tumor segmentation) [1, 3].
  - Collaborative learning for multi-modal **COVID-19 diagnosis** with X-ray and Ultrasound imagery [3].
  - Detecting **adverse events in mass-scale vaccination programs** [3].
  - Analysis of biomedical data (e.g., subcordinal brain changes in neurological disease) [1].
  - Drug discovery (e.g., MELLODDY project) [3].
  - Training on **multi-institutional datasets** constrained by HIPAA/FERPA [4].
- **Finance: Finance risk prediction for reinsurance** [4]. Collaborative training for **fraud detection** models [4]. Used by Webank and Swiss Re for high-precise financial analysis [3].

- **Recommender Systems:** Generating personalized recommendations using collaborative filtering [1].
- **Online Retailers:** Analyzing user click-stream data to enhance prediction of consumer's next browsing activities [1].
- **Model Types:** Applicable to models that operate over large inventories where clients interact with only a tiny fraction of items (sparse problems), such as **natural language models** or **content ranking models** using an embedding lookup table [4].

Advantages

FL provides significant advantages over traditional centralized and on-device machine learning models, primarily related to privacy, cost, and system efficiency.

- **Data Privacy and Security Guarantees:** FL is a **privacy-preserving decentralized approach** that keeps **raw data on-devices** and precludes direct access to it [1]. The approach involves sharing focused model updates (e.g., gradients) instead of the raw data [2]. Local training preserves the privacy, confidentiality, and integrity of patient data [3].
- **Compliance and Risk Mitigation:** FL can **mitigate many of the systemic privacy risks and costs** resulting from traditional centralized approaches [4]. It effectively handles the tradeoff between model learning and regulatory compliance (e.g., HIPAA/GDPR) because raw data is not centralized [3].
- **Communication and Network Efficiency:** FL eliminates the data communication overhead associated with centralized systems [1]. By pushing computation to the edge and requiring only small, iterative model updates, it reduces strain on the network [2].
- **Knowledge Sharing (vs. On-Device ML):** It overcomes the limitation of isolated on-device ML (where models don't benefit from peers' data) by aggregating local models to **share knowledge** among participants [1].
- **Accuracy and Model Diversity:** FL achieves **high precision and accuracy** by leveraging a large volume of data across many clients [1]. It also enables training on **multi-institutional datasets** (e.g., medical, education) where centralization was constrained, potentially leading to improved model **fairness** and diversity [4].
- **Resource and Latency Reduction:** FL handles the challenge of expensive centralized training [3]. It resolves the network latency problem as clients process data locally, eliminating the need to fetch data from a remote server [1].

Limitations: Challenges and Open Research Problems

FL faces significant technical and theoretical challenges spanning multiple domains.

Statistical and Optimization Challenges

- **Non-IID and Unbalanced Data:** Client data is typically **non-independent and non-identically distributed (Non-IID)** and **unbalanced** across clients [1, 2]. This causes standard algorithms like FedAvg to diverge or perform poorly [1].
- **Optimization Difficulty:** Local steps (K¿1) in FedAvg may slow convergence, and heterogeneous local steps can cause algorithms to converge to **mismatched objective functions** [4].
- **ML Workflow Adaptation:** Adapting centralized workflows (hyperparameter tuning, neural architecture search, debugging) to decentralized FL settings presents major challenges [4].

System and Infrastructure Challenges

- **Communication Bottleneck:** Many communication rounds and slow uplink connections create major bottlenecks [1].
- **Systems Heterogeneity:** Heterogeneous hardware, network connections, and power budgets cause **longer training times** and high **client dropout rates** [1, 2].
- **System Induced Bias:** Client selection based on charging status or Wi-Fi availability introduces **selection bias** [4].

Privacy, Security, and Robustness Challenges

- **Model Update Leakage:** Gradients can reveal sensitive client information [1].
- **Adversarial Attacks:** FL is vulnerable to **poisoning attacks** (model update poisoning, data poisoning) and **backdoor attacks** [1].
- **Privacy/Utility Trade-offs: Differential Privacy** degrades accuracy [1], while **cryptographic methods** (SMPC/HE) impose high computational costs [1, 3].
- **Evolving Data Privacy:** Updating models on **dynamically evolving databases** while maintaining privacy guarantees remains an open challenge [4].

Paper Titles Cited:

1. A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond
2. Federated Learning: Challenges, Methods, and Future Directions
3. Adoption of Federated Learning for Healthcare Informatics: Emerging Applications and Future Directions
4. Advances and Open Problems in Federated Learning