MODEL AGGREGATION PRINCIPLES IN FEDERATED LEARNING

Evgeny Osipov

LULEÅ UNIVERSITY OF TECHNOLOGY

Introduction to Federated Learning

Federated Learning (FL) is a distributed machine learning paradigm that enables training models across multiple decentralized devices or servers while keeping data localized.

Key Characteristics:

- Privacy Preservation: Training data remains on local devices
- Decentralized Architecture: No central data repository
- Collaborative Training: Multiple parties contribute to model improvement
- Communication Efficiency: Only model updates are exchanged

LULEÅ UNIVERSITY OF TECHNOLOGY

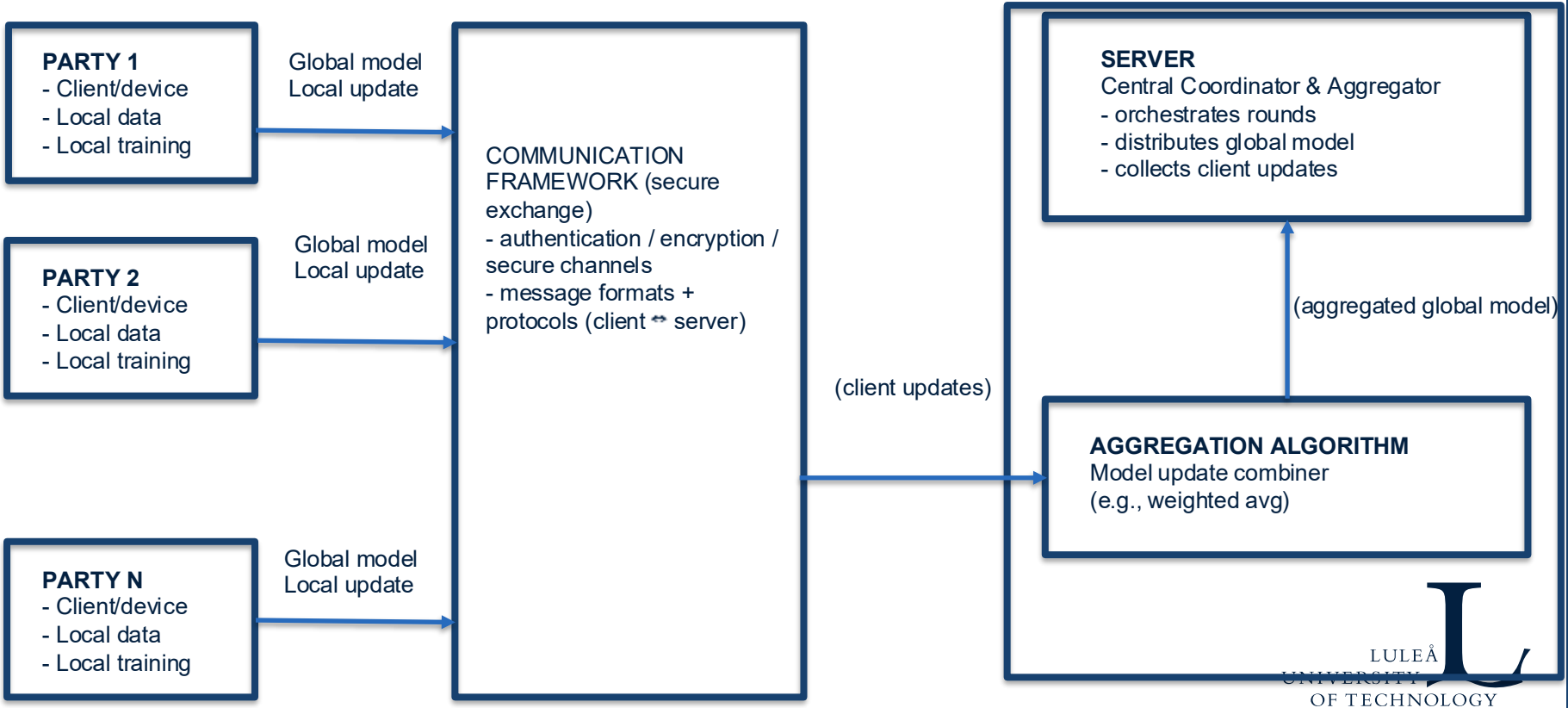
Architecture Overview: Four Core Components

Federated Learning systems consist of four fundamental architectural components.

Core Components:

1. **Server**: Central coordinator and aggregator
2. **Parties**: Distributed clients/participants
3. **Communication Framework**: Secure data exchange protocols
4. **Aggregation Algorithm**: Model update combination logic

LULEÅ UNIVERSITY OF TECHNOLOGY


PARTY 1
- Client/device
- Local data
- Local training
Global model
Local update
PARTY 2
- Client/device
- Local data
- Local training
Global model
Local update
PARTY N
- Client/device
- Local data
- Local training
Global model
Local update
COMMUNICATION FRAMEWORK (secure exchange)
- authentication / encryption / secure channels
- message formats + protocols (client ↔ server)
(client updates)
SERVER
Central Coordinator & Aggregator
- orchestrates rounds
- distributes global model
- collects client updates
(aggregated global model)
AGGREGATION ALGORITHM
Model update combiner
(e.g., weighted avg)

Architecture Overview: Server

The server orchestrates the FL process, acting as the primary point of control and aggregation.

Primary Functions:

- Client selection and round management
- Parameter distribution to participating clients
- Model aggregation from client updates
- Performance monitoring and compliance

SERVER
Central Coordinator & Aggregator
- orchestrates rounds
- distributes global model
- collects client updates

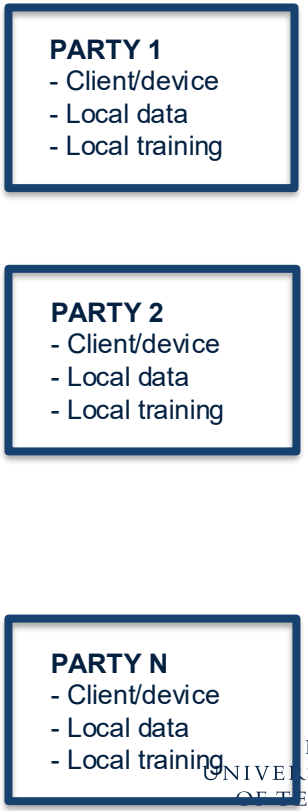
LULEÅ UNIVERSITY OF TECHNOLOGY

Architecture Overview: Parties

Parties are the distributed entities/clients that own training data and perform local model training.

Two Main Categories:

- **Cross-Device FL**: Mobile devices, IoT sensors
- **Cross-Silo FL**: Organizations, companies, institutions

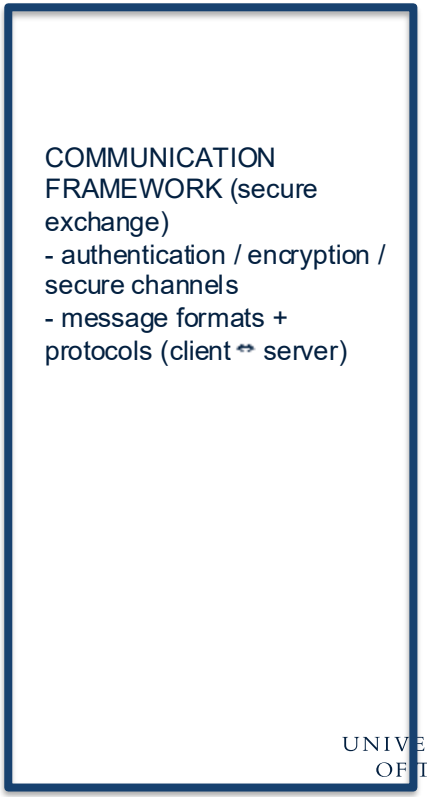


Architecture Overview: Communication framework

The communication framework encompasses protocols for secure, efficient data exchange.

Core Layers:

- **Transport**: TCP/IP, UDP, WebSockets
- **Security**: TLS/SSL, certificates
- **Application**: gRPC, REST APIs, MQTT



FL Categorization Overview

Federated Learning can be categorized based on data distribution patterns across participants:

1. Horizontal FL (HFL):

- Same feature spaces, different samples
- Most common form (FedAvg, cross-device)
- Scales to millions of participants

2. Vertical FL (VFL):

- Different feature spaces, overlapping samples
- Feature-partitioned data across organizations
- Requires cryptographic techniques

3. Federated Transfer Learning:

- Combines FL with transfer learning
- Domain adaptation across heterogeneous environments
- Addresses cold start and data scarcity

LULEÅ UNIVERSITY OF TECHNOLOGY

Horizontal Federated Learning (HFL)

Horizontal Federated Learning occurs when participants have identical data schemas but different samples.

Key Characteristics:

- Feature Space: Same attributes/columns across all participants
- Sample Space: Different/non-overlapping data samples
- Scalability: Naturally scales to large numbers of participants
- Privacy: Strong guarantees with minimal data overlap

FedAvg Algorithm:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1}$$

LULEÅ UNIVERSITY OF TECHNOLOGY

HFL Use Cases

HFL powers many real-world applications.

Cross-Device FL:

- Gboard: Next-word prediction across Android devices
  - https://support.google.com/gboard/answer/12373137?hl=en
- Mobile Apps: Personalized recommendations
- IoT Networks: Smart device coordination

Cross-Silo FL:

- Banks: Fraud detection collaboration
- Hospitals: Medical research without data sharing
- Companies: Enterprise model training

LULEÅ UNIVERSITY OF TECHNOLOGY

HFL Challenges

HFL faces challenges from participant and data diversity.

Main Challenges:

- Statistical heterogeneity (non-IID data)
- System heterogeneity (varying compute)
- Communication costs and client dropout

LULEÅ UNIVERSITY OF TECHNOLOGY

Vertical Federated Learning (VFL)

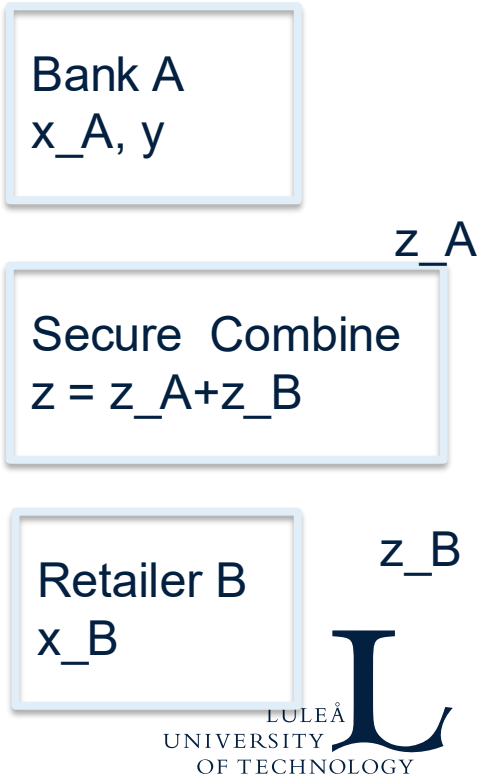
Vertical Federated Learning occurs when participants have different features for the same entities.

Key Characteristics:

- Feature Space: Different attributes across participants
- Sample Space: Overlapping users/entities
- Privacy: Very strong (cryptographic techniques required)
- Coordination: Complex sample alignment needed

Feature Partitioning:

$$x_i = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(K)}]$$



LULEÅ UNIVERSITY OF TECHNOLOGY

VFL Use Cases

VFL enables collaboration across different data silos.

Financial Services:

- Banks share features for credit scoring
- Transaction history + employment data + credit history
- Better risk assessment without data sharing

Healthcare:

- Hospitals combine different medical data
- Lab results + imaging + patient history
- Improved diagnosis with comprehensive data

LULEÅ UNIVERSITY OF TECHNOLOGY

VFL Challenges

VFL requires sophisticated cryptographic techniques.

Main Challenges:

- Sample alignment without sharing identifiers
- Secure multi-party computation
- Homomorphic encryption for joint operations

Solutions:

- Secret Sharing: Distribute computations
- Homomorphic Encryption: Encrypted arithmetic
- Secure MPC: Privacy-preserving protocols

LULEÅ UNIVERSITY OF TECHNOLOGY

Federated Transfer Learning

Type	Users	Features	Example
Horizontal FL	Different	Same	Hospitals in different cities
Vertical FL	Same	Different	Bank + Retail
Federated Transfer Learning	**Different**	**Different**	Hospital + Insurance

LULEÅ UNIVERSITY OF TECHNOLOGY

Federated Transfer Learning

FL + Transfer Learning Combines federated learning with transfer learning techniques for domain adaptation.

Key Characteristics:

- Raw data never shared
- Only encrypted embeddings exchanged
- Overlapping ID matching done securely
- Often uses:
  - Secure Multi-Party Computation
  - Homomorphic encryption
  - Differential privacy

LULEÅ UNIVERSITY OF TECHNOLOGY

Architectural view

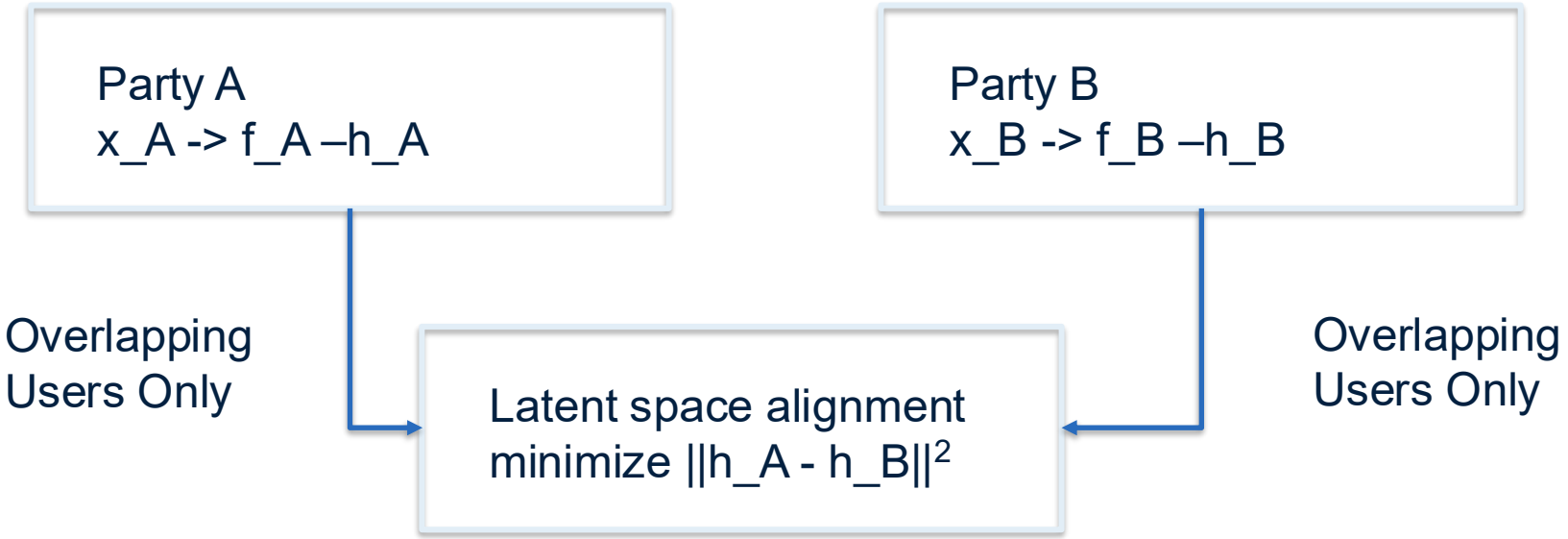
Party A
x_A -> f_A –h_A

Party B
x_B -> f_B –h_B

- x_A, x_B– features (in general non-overlapping)
- f_A – locally trained model to solve organization-specific task
  - E.g. decease forecasting
- h_A – learned feature representation

LULEÅ UNIVERSITY OF TECHNOLOGY

Architectural view



After alignment (Even for non-overlapping users):

- Party A (e.g. hospital) model benefits from structural patterns learned by Party B (e.g. Insurer).
- Party B helps shape the embedding space of Party A for better generalization

LULEÅ UNIVERSITY OF TECHNOLOGY

Summary so far: FL Categories Comparison

Use HFL when:

- Large number of participants with similar data structures
- Cross-device applications (mobile, IoT)
- Strong privacy with minimal data overlap

Use VFL when:

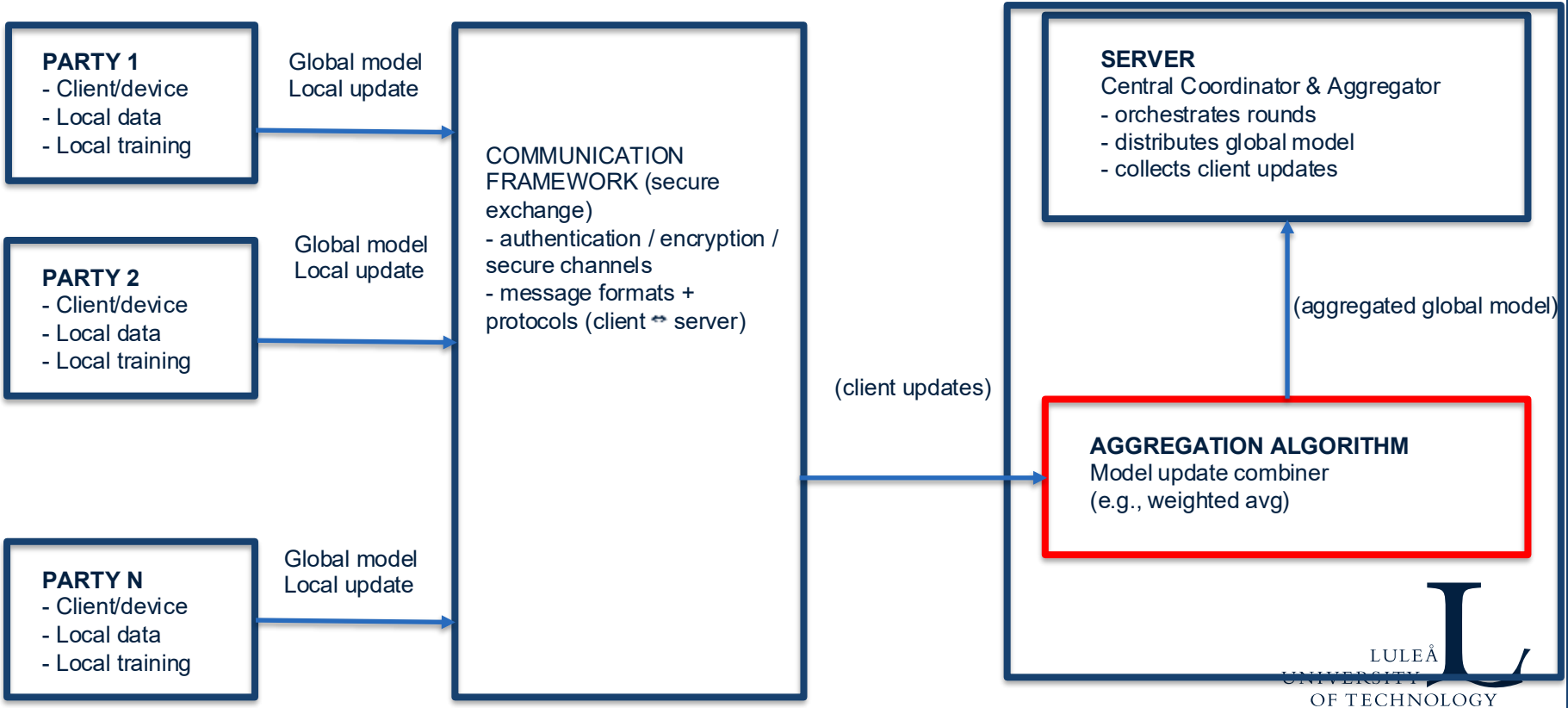
- Few participants with complementary features
- Sensitive data requiring strong privacy guarantees
- Financial services, healthcare applications

Use Federated Transfer Learning when:

- Participants with different but related domains
- Cold start problems with limited local data
- Evolving environments requiring continual adaptation
- Multi-task learning scenarios

LULEÅ UNIVERSITY OF TECHNOLOGY

Recall. What is left?

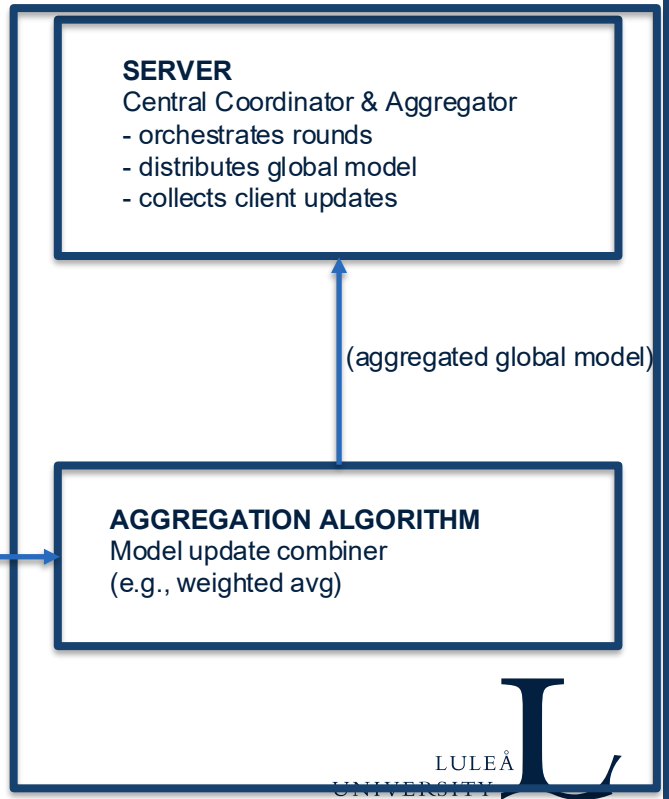


What is Model Aggregation - HFL?

Model aggregation is the process of combining individual client models to create a global model in federated learning.

Goals:

- Leverage collective knowledge from distributed clients
- Maintain data privacy (no raw data exchange)
- Handle client heterogeneity (data, compute, network)
- Ensure robustness against malicious participants
- Optimize for communication efficiency



LULEÅ UNIVERSITY OF TECHNOLOGY

Federated Averaging (FedAvg)

Mathematical Formulation:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1}$$

Where:

- w_k^{t+1}: Client k's model after local training
- n_k: Data samples on client k
- $n = \sum n_k$: Total samples across clients

LULEÅ UNIVERSITY OF TECHNOLOGY

Example FedAvg: Round 1

Task: Predict disease risk (binary classification)

Model: Single parameter w (for simplicity)

Initial Model **Server initializes:** $w^{(0)} = 0.5$

Step 1: Server sends $w^{(0)} = 0.5$ to all clients

Step 2: Each client trains locally on their data

Step 3: After local training (e.g., 5 epochs of SGD) clients send their trained models back to server

LULEÅ UNIVERSITY OF TECHNOLOGY

Example FedAvg: Round 1 (cont.)

Step 4: Server computes weighted average

Formula:

$$w^{(1)} = \frac{n_A}{n} \cdot w_A^{(1)} + \frac{n_B}{n} \cdot w_B^{(1)} + \frac{n_C}{n} \cdot w_C^{(1)}$$

Substituting values:

$$\begin{aligned} w^{(1)} &= \frac{100}{500} \cdot 0.65 + \frac{200}{500} \cdot 0.80 + \frac{200}{500} \cdot 0.75 \\ &= 0.20 \cdot 0.65 + 0.40 \cdot 0.80 + 0.40 \cdot 0.75 \\ &= 0.13 + 0.32 + 0.30 = \mathbf{0.75} \end{aligned}$$

New global model: $w^{(1)} = 0.75$

LULEÅ UNIVERSITY OF TECHNOLOGY

Example FedAvg: Round 1 (cont.)

Compare:

- Hospital A: $w_A = 0.65$) weight: 0.20)
- Hospital B: $w_B = 0.80$) weight: 0.40)
- Hospital C: $w_C = 0.75$) weight: 0.40)
- **Global:** $w^{(1)} = 0.75$

Observe:

- Result is closer to B and C (they have more data)
- Hospital A has less influence (only 20% of data)
- Simple average would give: $(0.65 + 0.80 + 0.75)/3 = 0.733$
- Weighted average: 0.75 (accounts for data size)

LULEÅ UNIVERSITY OF TECHNOLOGY

Example FedAvg: Round 2 and so on

Step 1: Server sends new global model $w^{(1)} = 0.75$ to all clients

Step 2: Each client trains locally starting from $w^{(1)} = 0.75$

After local training:

Step 3: Send updates to server

Step 4: Server aggregates:

$$w^{(2)} = 0.20 \cdot 0.78 + 0.40 \cdot 0.82 + 0.40 \cdot 0.80$$
$$= 0.156 + 0.328 + 0.320 = \mathbf{0.804}$$

Why weighted is better:

- Hospitals B and C have 4× more data than A
- Their models are trained on more samples (more reliable)
- Weighted average gives them appropriate influence
- Leads to better convergence and final model quality

LULEÅ UNIVERSITY OF TECHNOLOGY

FL Security Challenges

Federated Learning introduces distinctive security challenges due to its distributed nature and privacy-preserving design.

Security threats in FL can undermine both privacy and utility!

LULEÅ UNIVERSITY OF TECHNOLOGY

Attack Impact

Poisoning attacks have cascading effects on FL systems and stakeholders.

Individual Model Level:

- Incorrect feature engineering
- Regulatory compliance failures
- Debugging based on false explanations

System-Level Impact:

- Trust erosion in AI systems
- Economic consequences from poor decisions
- Hidden security risks in explanations

Societal Impact:

- Bias amplification or masking
- Accountability issues
- Flawed policy decisions

LULEÅ UNIVERSITY OF TECHNOLOGY

Attack Variants

Attacks can vary based on attacker resources and coordination.

Single-Client Attacks:

- Limited impact but hard to detect
- Blends with natural data variations

Multi-Client Coordinated Attacks:

- Sybil attacks with multiple identities
- Combined effect through FedAvg
- Higher impact but easier to detect

Stealthy Attacks:

- Minimal perturbations
- Adaptive poisoning strategies
- Integration with backdoor attacks

LULEÅ UNIVERSITY OF TECHNOLOGY

FL Threat Taxonomy

FL security threats can be categorized into three primary areas.

1. **Privacy Attacks**: Information leakage from model outputs
2. **Poisoning Attacks**: Malicious data or model manipulation
3. **Fairness-Robustness Trade-offs: Balancing multiple objectives**
4. **Utility-Robustness Trade-offs**

Each category presents unique challenges for FL system design!

LULEÅ UNIVERSITY OF TECHNOLOGY

Privacy Risks

Even without sharing raw data, FL systems can leak sensitive information through model outputs and explanations.

Key Privacy Threats:

- **Membership Inference**: Determine if sample was in training data
- **Property Inference**: Learn statistical properties of training data
- **Explanation Leakage**: Sensitive information through model explanations

LULEÅ UNIVERSITY OF TECHNOLOGY

Differential Privacy in FL

Add noise to model updates to provide provable privacy protection.

Local DP: Each client adds noise before sending:

$$\widetilde{w}_k = w_k + \mathcal{N}(0, \sigma^2)$$

Central DP: Server adds noise to aggregated updates:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t + \mathcal{N}(0, \sigma^2)$$

LULEÅ UNIVERSITY OF TECHNOLOGY

Poisoning Attacks: Overview

Poisoning attacks involve malicious participants introducing corrupted data or model updates into the FL training process.

Attack Types:

- **Model Poisoning**: Send manipulated model updates
- **Data Poisoning**: Manipulate local training data
- **Byzantine Attacks**: Arbitrary malicious behavior

Poisoning can target accuracy, explanations, or information extraction!

LULEÅ UNIVERSITY OF TECHNOLOGY

Byzantine Threat

A **Byzantine** participant can behave **arbitrarily**:

- send incorrect or inconsistent messages,
- send extreme/malicious gradients/updates,
- collude with others, or remain silent.

In FL, Byzantine clients can:

- break convergence,
- insert backdoors,
- dominate averaging with outliers.

LULEÅ UNIVERSITY OF TECHNOLOGY

Model poisoning

A **model poisoning attack** occurs when a malicious client:

- Intentionally modifies its local update
- To manipulate the global model
- While still following the communication protocol

Different from data poisoning (corrupting local data)

LULEÅ UNIVERSITY OF TECHNOLOGY

Why FedAvg is Vulnerable

FedAvg assumes:

- Clients are honest.
- Updates reflect legitimate local training.
- Averaging is safe.

But averaging has **no robustness mechanism**.

So a malicious update can:

- Shift the global model
- Inject a backdoor
- Prevent convergence
- Reduce accuracy

LULEÅ UNIVERSITY OF TECHNOLOGY

Attacks against FedAvg

With FedAvg:

$$w_{t+1} = w_t + \frac{1}{n}\sum_{i=1}^{n} \Delta_i \,.$$

If one client m is malicious:

$$w_{t+1} = w_t + \frac{1}{n}\left(\sum_{i \neq m} \Delta_i + \Delta_m\right).$$

A malicious client can scale its update:

$$\Delta_m = n \cdot \Delta^\star \;\Rightarrow\; w_{t+1} \approx w_t + \Delta^\star$$

so a **single client can dominate the round** (“model replacement” style effect).

LULEÅ UNIVERSITY OF TECHNOLOGY

Model Poisoning: Common Attack Styles

- **Sign-flip / disruption:** $\Delta_m = -\alpha\,\Delta_{\text{honest}}$ to slow or reverse learning.
- **Backdoor:** train locally so that a trigger T causes prediction $\hat{y} = y^\star$ while keeping normal accuracy high.
- **Model replacement:** craft Δ_m to push w_{t+1} close to an attacker-chosen $w^\star$

LULEÅ UNIVERSITY OF TECHNOLOGY

Data Poisoning Against Interpretations

A sophisticated attack that undermines model interpretability by poisoning training data to alter feature importance attributions.

Core Insight:

Model interpretability methods are sensitive to small, carefully crafted perturbations in training data.

Key Difference:

- Traditional attacks target model **accuracy**
- This attack targets model **explanations**
- Maintains accuracy while corrupting interpretability

LULEÅ UNIVERSITY OF TECHNOLOGY

Attack Methodology

Four-Step Poisoning Process Systematic approach to manipulate model explanations in FL.

Step 1 - Target Selection: Choose explanation method (SHAP, LIME, feature importance)

Step 2 - Poison Generation: Solve optimization/or craft to create poisoned samples

Step 3 - Explanation Manipulation: Feature/Label flip, masking, or inversion attacks

Step 4 - FL Integration: Distribute poison across clients, consider FedAvg effects

LULEÅ UNIVERSITY OF TECHNOLOGY

Mathematical Formulation

Objective Function:

$$\min_{\delta} \| \phi(f(\theta, x + \delta)) - \phi_{target} \|^2$$

Constraints:

$$\| \delta \| \leq \epsilon \quad \text{(perturbation budget)}$$

$$\mathcal{L}(f(\theta, x + \delta), y) \leq \tau \quad \text{(accuracy preservation)}$$

Trade-off between explanation manipulation and accuracy preservation!

LULEÅ UNIVERSITY OF TECHNOLOGY

Manipulation Strategies

Three Attack Variants Different approaches to corrupt model interpretability.

1. Feature Flip Attacks:

- Make irrelevant features appear important
- Example: Credit model prioritizes "favorite color" over "credit history"

2. Feature Masking Attacks:

- Hide importance of critical features
- Example: Reduce apparent biomarker importance in medical diagnosis

3. Explanation Inversion Attacks:

- Make explanations contradict true decision process
- Example: Income-based model appears to prioritize education

LULEÅ UNIVERSITY OF TECHNOLOGY

Defense Strategies

FL defense strategies address different attack types with appropriate countermeasures.

Defense Categories:

1. **Robust Aggregation**: Defend against malicious updates
2. **Client Authentication**: Verify participant legitimacy
3. **Data Validation**: Detect poisoned training data
4. **Privacy Protection**: Prevent information leakage
5. **Fairness Mechanisms**: Ensure equitable performance

LULEÅ UNIVERSITY OF TECHNOLOGY

Detection and Mitigation

Statistical Detection:

- Explanation consistency checks
- Outlier detection in explanation patterns
- Distribution shift monitoring

Robust Explanation Methods:

- Ensemble explanations
- Certified explanations with guarantees
- Adversarial training for explanations

LULEÅ UNIVERSITY OF TECHNOLOGY

Robust Aggregation Methods: Coordinate-wise Median

Principle: Take median of each parameter coordinate across all client updates.

Formula:

$$w_i^{t+1} = \text{median}(\{w_{k,i}^{t+1}\}_{k=1}^{K})$$

Advantages:

- Robust to value manipulation attacks
- Computationally simple
- No hyperparameters to tune

LULEÅ UNIVERSITY OF TECHNOLOGY

Robust Aggregation Methods: Krum Aggregation

Principle Select the update that is closest to other (gradient) updates (multi-dimensional median).

Algorithm:

1. For each client k, compute $\sum_{k' \neq k} \| g_k - g_{k'} \|^2$,where
2. Select k^* with minimum distances
3. Average top m closest updates

g's are exchanged gradients

LULEÅ UNIVERSITY OF TECHNOLOGY

Robust Aggregation Methods: Krum Aggregation

Setup

- Number of clients: n=6
- Max Byzantine clients: f=1
- Each client sends a **model update vector g**.
- Krum uses **squared Euclidean distances**.

Krum parameter:

- m=n−f−2 = 6−1−2 = 3

- So for each update gi, we find its **3 nearest neighbors** (by squared distance) and sum those distances. The update with the **smallest sum** is selected.

LULEÅ UNIVERSITY OF TECHNOLOGY

Robust Aggregation Methods: Krum Aggregation

- Honest updates (close to each other):
  - g1=[1.00,1.00], g2=[1.10,0.90], g3=[0.90,1.20], g4=[1.05,1.05], g5=[0.95,0.85]
- Byzantine (malicious outlier):
  - g6=[10,−10]

LULEÅ UNIVERSITY OF TECHNOLOGY

Robust Aggregation Methods: Krum Aggregation

Use updates from these nodes

	g1	g2	g3	g4	g5	g6
g1	0	0.020	0.050	0.005	0.025	202.000
g2	0.020	0	0.130	0.025	0.025	198.020
g3	0.050	0.130	0	0.045	0.125	208.250
g4	0.005	0.025	0.045	0	0.050	202.205
g5	0.025	0.025	0.125	0.050	0	199.625
g6	202.000	198.020	208.250	202.205	199.625	0

LULEÅ UNIVERSITY OF TECHNOLOGY

Fairness-Robustness Trade-offs

In FL:

- Clients often have **non-IID data**.
- Some clients represent minority populations.
- Their updates may naturally look different from the majority.

Robust aggregation methods (e.g., Krum, trimmed mean, median):

- Treat large deviations as suspicious.
- Remove or downweight outliers.
- But minority updates can look like outliers even if honest.

LULEÅ UNIVERSITY OF TECHNOLOGY

Fairness-Robustness Trade-offs

Joint Optimization Approaches Methods to achieve both robustness and fairness simultaneously.

Fairness-aware FL may require:

- Personalized models
- Reweighting
- Clustering
- Multi-task learning
- Adaptive thresholds

Proposed Solutions:

- Multi-objective optimization frameworks
- Fairness-aware adversarial training
- Regularization techniques for balanced performance

Multi-objective optimization needed for both robustness and fairness!

LULEÅ UNIVERSITY OF TECHNOLOGY

Secure federated learning: Secure Aggregation

Secure aggregation ensures the server can compute the aggregate of client updates without learning anything about individual client contributions.

Security Requirements:

1. Privacy: Server learns only the aggregate
2. Correctness: Honest inputs included correctly
3. Robustness: Works with malicious clients
4. Efficiency: Reasonable computation/communication

LULEÅ UNIVERSITY OF TECHNOLOGY

Homomorphic Encryption:principle

Homomorphic encryption allows computation on encrypted data without decryption.

Key Property (Additive HE):

$$Enc(a) \cdot Enc(b) = Enc(a + b)$$

Schemes:

- Paillier: Probabilistic, additive homomorphic
- CKKS/BFV: Support floating-point operations

Google if you want to know more: homomorphuic encryption is an advanced topic in cryptography

LULEÅ UNIVERSITY OF TECHNOLOGY

Homomorphic Encryption:principle

SecureAgg Protocol (Google) uses threshold homomorphic encryption in TensorFlow Federated.

4-Phase Protocol:

1. Setup: Clients generate keys, establish secure channels
2. Masking: Clients add random masks to updates
3. Aggregation: Server sums encrypted masked updates
4. Unmasking: Clients reveal masks to cancel randomness

LULEÅ UNIVERSITY OF TECHNOLOGY

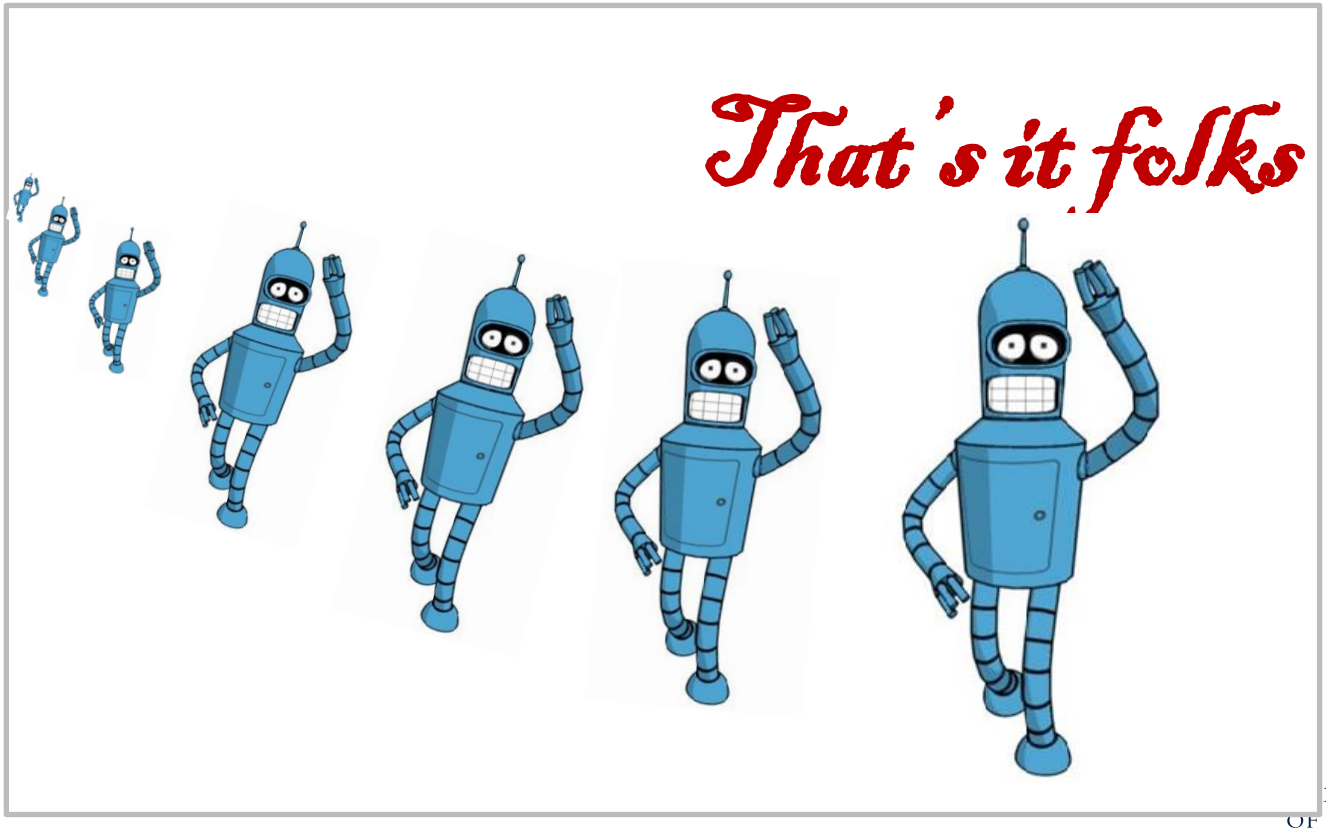
Homomorphic Encryption:Paillier Cryptosystem

- Key Gen: (n, g) where $n = p \cdot q$
- Encryption: $Enc(x) = g^x \cdot r^n \bmod n^2$
- Addition: $Enc(x) \cdot Enc(y) = Enc(x + y)$

FL Protocol:

1. Server generates Paillier key pair (pk, sk)
2. Clients: $c_k = Enc_{pk}(w_k)$
3. Server: $C = \prod c_k = Enc_{pk}(\sum w_k)$
4. Server: $\sum w_k = Dec_{sk}(C)$

LULEÅ UNIVERSITY OF TECHNOLOGY



LULEÅ UNIVERSITY OF TECHNOLOGY