Supplementary Material: FL-EHDS: A Privacy-Preserving Federated Learning Framework for the European Health Data Space

Fabio Liberti
Department of Computer Science
Universitas Mercatorum, Rome, Italy
fabio.liberti@unimercatorum.it
ORCID: 0000-0003-3019-5411

***Abstract*—This document provides supplementary material for the FL-EHDS paper, including complete algorithm pseudocode for all framework components, extended experimental figures, detailed algorithm comparison analysis, advanced FL paradigm descriptions, infrastructure component specifications, extended EHDS interoperability details, and clinical imaging experiment configurations. The open-source reference implementation (∼40K lines, 159 modules) is available at https://github.com/FabioLiberti/FL-EHDS-FLICS2026.**

I. Algorithm Pseudocode

This section provides formal algorithmic descriptions of all FL-EHDS framework components, demonstrating EHDS governance integration at every layer.

A. FedAvg with EHDS Compliance

Algorithm S1 presents the core federated averaging procedure adapted for EHDS regulatory requirements, operating in a client-server architecture where the central aggregator coordinates training across distributed hospital nodes within a Secure Processing Environment.

Key Design Decisions:

- **ValidatePermit**: Before each round, the HDAB-issued permit is verified against temporal bounds and Article 53 permitted purposes.
- **SelectParticipants**: Configurable client selection—full participation or sampling for large federations.
- **FilterOptedOut**: Records from citizens who exercised Article 71 opt-out rights are excluded *before* gradient computation.
- **Weighted Aggregation**: Gradients weighted by local dataset size (n_h), following original FedAvg [1].
- **ClipGradient**: L2-norm clipping bounds individual contributions, providing sensitivity bounds for DP.

B. Gaussian Differential Privacy Mechanism

Algorithm S2 implements the Gaussian mechanism for differential privacy, applied at the aggregation server after receiving clipped gradients.

```
Algorithm S1: FL-EHDS FedAvg Training
Input: Hospitals $\mathcal{H} = \{h_1, \ldots, h_K\}$, permit $P$, rounds $T$
Output: Global model $\theta^{(T)}$
Server executes:
    Initialize $\theta^{(0)}$
    for round $t = 1$ to $T$ do
        // Governance check (Layer 1)
        if not ValidatePermit($P$, $t$) then abort
        $\mathcal{H}_t \leftarrow$ SelectParticipants($\mathcal{H}$)
        for each hospital $h \in \mathcal{H}_t$ in parallel do
            $\Delta_h^{(t)}, n_h \leftarrow$ LocalTrain($h$, $\theta^{(t-1)}$)
        // Aggregation with privacy (Layer 2)
        $\theta^{(t)} \leftarrow \theta^{(t-1)} + \frac{1}{\sum_h n_h} \sum_{h \in \mathcal{H}_t} n_h \cdot \Delta_h^{(t)}$
        LogCompliance($t$, $\mathcal{H}_t$)
    return $\theta^{(T)}$
LocalTrain($h$, $\theta$) at hospital $h$:
    // Opt-out filtering (Article 71)
    $\mathcal{D}_h \leftarrow$ FilterOptedOut($\mathcal{D}_h$, OptOutRegistry)
    $\theta_h \leftarrow \theta$
    for epoch $e = 1$ to $E$ do
        for batch $\mathcal{B} \in \mathcal{D}_h$ do
            $\theta_h \leftarrow \theta_h - \eta \nabla \mathcal{L}(\theta_h; \mathcal{B})$
    $\Delta_h \leftarrow \theta_h - \theta$
    // Privacy protection (Layer 3)
    $\Delta_h \leftarrow$ ClipGradient($\Delta_h$, $C$)
    return $\Delta_h$, $|\mathcal{D}_h|$
```

Mathematical Foundation: The noise scale $\sigma = C \cdot \sqrt{2\ln(1.25/\delta)}/\varepsilon$ guarantees (ε, δ)-DP. The cumulative privacy expenditure is tracked using Rényi DP (RDP) [6] composition, providing 5–6× tighter bounds than naive composition.

Practical Considerations:

- $\varepsilon = 10$: moderate noise, ∼5.2pp accuracy drop
- $\varepsilon = 1$: strong privacy, ∼5.8pp accuracy drop
- The ε selection must be negotiated with HDABs during permit approval

C. HDAB Permit Validation

Algorithm S3 ensures all FL operations comply with the data permit issued by HDABs, implementing EHDS Articles 53 and GDPR Article 30.

```
Algorithm S2: Gaussian DP Mechanism
Input: Gradient $\Delta$, sensitivity $C$, privacy budget $\varepsilon$, $\delta$
Output: Noisy gradient $\tilde{\Delta}$
// Compute noise scale from Gaussian mechanism
$\sigma \leftarrow C \cdot \sqrt{2\ln(1.25/\delta)}/\varepsilon$
// Add calibrated Gaussian noise to each parameter
for each parameter $w \in \Delta$ do
    $\tilde{w} \leftarrow w + \mathcal{N}(0, \sigma^2)$
// Track cumulative privacy expenditure
PrivacyAccountant.spend($\varepsilon$)
if PrivacyAccountant.budget_exhausted() then
    raise PrivacyBudgetExhaustedError
return $\tilde{\Delta}$
```

```
Algorithm S3: Data Permit Validation
Input: Permit $P$, round $t$, requested categories $\mathcal{C}$
Output: Boolean validity
// Check temporal validity
if CurrentTime() $>$ $P$.valid_until then
    raise PermitExpiredError
// Check purpose alignment (Article 53)
if $P$.purpose $\notin$ AllowedPurposes then
    raise PurposeMismatchError
// Check data category authorization
for each category $c \in \mathcal{C}$ do
    if $c \notin P$.authorized_categories then
        raise UnauthorizedCategoryError
// Log access for GDPR Article 30
AuditTrail.log(permit=$P$, round=$t$, categories=$\mathcal{C}$)
return True
```

D. Secure Aggregation Protocol

Algorithm S4 implements secure aggregation using Shamir's secret sharing and pairwise masking, ensuring the server observes only the aggregate gradient.

Protocol Phases: (1) Each client splits gradients into K shares using (t, K)-threshold Shamir secret sharing; (2) Clients add pairwise random masks negotiated via ECDH key exchange; (3) The server computes the sum—masks cancel out and only the true aggregate remains.

```
Algorithm S4: Secure Aggregation (Pairwise Masking)
Input: Client gradients $\{\Delta_1, \ldots, \Delta_K\}$, threshold $t$
Output: Aggregated gradient $\Delta_{agg}$
// Phase 1: ECDH key exchange + Shamir sharing
for each client $k$ do
    shares$_k$ $\leftarrow$ ShamirShare($\Delta_k$, $t$, $K$)
    Distribute shares$_k$ to other clients
// Phase 2: Add pairwise random masks
for each client $k$ do
    $\hat{\Delta}_k \leftarrow \Delta_k + \sum_{j<k} r_{jk} - \sum_{j>k} r_{kj}$
// Phase 3: Server reconstructs aggregate
$\Delta_{agg} \leftarrow \sum_{k=1}^{K} \hat{\Delta}_k$
// Masks cancel: $\sum_k \sum_{j<k} r_{jk} - \sum_k \sum_{j>k} r_{kj} = 0$
if ActiveClients $< t$ then
    raise SecureAggregationError
return $\Delta_{agg}$
```

E. FedProx for Non-IID Data

Algorithm S5 extends FedAvg with a proximal term preventing local drift when hospitals have skewed populations [2].

```
Algorithm S5: FedProx Local Update
Input: Local data $\mathcal{D}_h$, global model $\theta$, proximal weight $\mu$
Output: Local update $\Delta_h$
$\theta_h \leftarrow \theta$
for epoch $e = 1$ to $E$ do
    for batch $\mathcal{B} \in \mathcal{D}_h$ do
        $g \leftarrow \nabla\mathcal{L}(\theta_h; \mathcal{B})$
        // Proximal term: $\nabla \frac{\mu}{2} \|\theta_h - \theta\|^2$
        $g \leftarrow g + \mu(\theta_h - \theta)$
        $\theta_h \leftarrow \theta_h - \eta \cdot g$
$\Delta_h \leftarrow \theta_h - \theta$
return $\Delta_h$
```

Parameter Selection: $\mu = 0$ reduces to FedAvg; $\mu \in [0.01, 0.1]$ provides stable convergence; $\mu > 1$ may prevent local adaptation.

F. Article 71 Opt-Out Registry Protocol

Algorithm S6 implements citizen opt-out enforcement with granular scope support.

```
Algorithm S6: Article 71 Opt-Out Filtering
Input: Local dataset $\mathcal{D}_h$, purpose $p$, categories $\mathcal{C}$
Output: Filtered dataset $\mathcal{D}'_h$
// Synchronize with national opt-out registry
OptOutRecords $\leftarrow$ FetchOptOutRegistry(MemberState)
$\mathcal{D}'_h \leftarrow \emptyset$
for each record $r \in \mathcal{D}_h$ do
    citizen_id $\leftarrow$ r.pseudonymized_id
    opted_out $\leftarrow$ False
    // Check purpose-specific opt-out
    if (citizen_id, $p$) $\in$ OptOutRecords then
        opted_out $\leftarrow$ True
    // Check category-specific opt-out
    for each $c \in \mathcal{C}$ do
        if (citizen_id, $c$) $\in$ OptOutRecords then
            opted_out $\leftarrow$ True
    if not opted_out then
        $\mathcal{D}'_h \leftarrow \mathcal{D}'_h \cup \{r\}$
// Log filtering statistics for audit
AuditLog.record(total=$|\mathcal{D}_h|$, filtered=$|\mathcal{D}'_h|$)
return $\mathcal{D}'_h$
```

G. FHIR R4 Preprocessing Pipeline

Algorithm S7 standardizes heterogeneous EHR data into FHIR R4 format. Given that only 34% of European providers achieve full FHIR compliance [7], this step is essential.

H. Privacy Budget Accountant

Algorithm S8 tracks cumulative privacy via Rényi DP moments, providing tighter bounds for multi-round training.

II. Supplementary Experimental Figures

This section presents detailed experimental results from the FL-EHDS benchmark suite. All figures are generated from real experimental runs available in the repository.

```
**Algorithm S7: FHIR R4 Preprocessing**
**Input:** Raw EHR records $\mathcal{R}$, feature specification $\mathcal{F}$
**Output:** Training tensors $(X, y)$
format ← DetectFormat($\mathcal{R}$) // HL7v2, CDA, CSV
parser ← GetParser(format)
records ← parser.parse($\mathcal{R}$)
// Map to standard terminologies
for each $r \in$ records do
    $r$.diagnoses ← MapToICD10($r$.diagnoses)
    $r$.medications ← MapToATC($r$.medications)
    $r$.labs ← MapToLOINC($r$.labs)
fhir_bundle ← ToFHIR(records)
ValidateFHIR(fhir_bundle)
$X$ ← ExtractFeatures(fhir_bundle, $\mathcal{F}$)
$X$ ← StandardScaler.fit_transform($X$)
$y$ ← ExtractLabels(fhir_bundle)
return $(X, y)$
```

```
**Algorithm S8: RDP Privacy Budget Accountant**
**Input:** Total budget $(\varepsilon_{total}, \delta_{total})$, rounds $T$
**Output:** Per-round budget allocation
$\lambda$ ← [0] × MAX_ORDER                         // Rényi moments
rounds_completed ← 0
function AllocateRound():
    $\varepsilon_{spent}$ ← ComputeEpsilon($\lambda$, $\delta_{total}$)
    $\varepsilon_{remaining} \leftarrow \varepsilon_{total} - \varepsilon_{spent}$
    if $\varepsilon_{remaining} < \varepsilon_{min}$ then
        raise BudgetExhaustedError
    $\varepsilon_t \leftarrow \varepsilon_{remaining}/(T-$ rounds_completed)
    return $\varepsilon_t$
function RecordRound($\sigma$, $q$):
    for order = 1 to MAX_ORDER do
        $\lambda$[order] += ComputeMoment(order, $\sigma$, $q$)
    rounds_completed += 1
```

A. Hospital Data Distribution

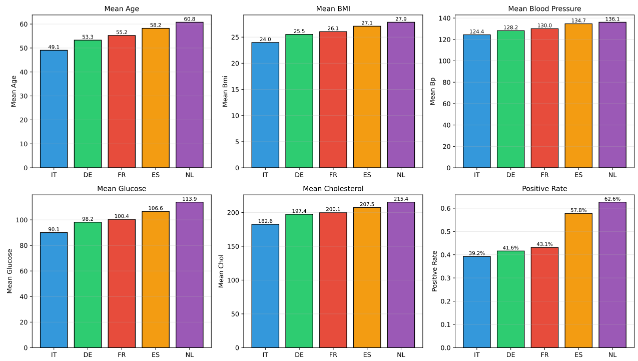

Fig. 1. Data distribution across hospitals. Notable heterogeneity: Amsterdam shows older population (60.8 years mean age) with higher positive rate (62.6%) compared to Rome (49.1 years, 39.2%). This reflects realistic cross-border EHDS variability.

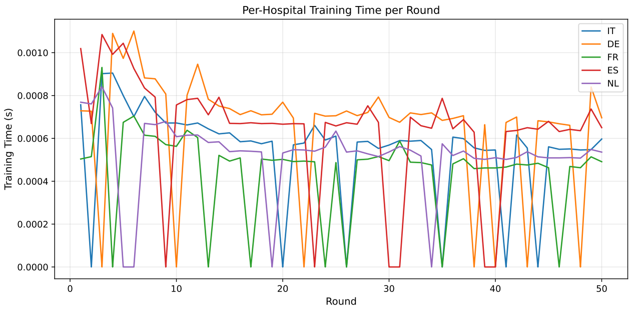

Fig. 2. Per-client training time per round. Larger hospitals (Berlin: 500 samples) exhibit slightly longer training times. The adaptive training engine compensates by adjusting batch sizes for stragglers.

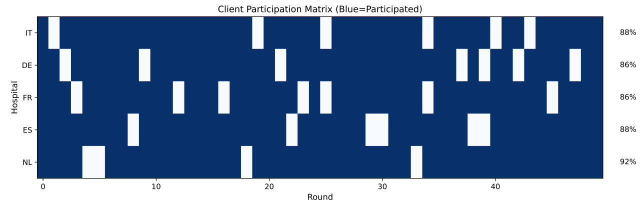

Fig. 3. Client participation matrix (50 rounds × 5 clients). Participation rates: IT 88%, DE 86%, FR 86%, ES 88%, NL 92%. The framework tolerates 10–15% dropout per round while maintaining convergence.

B. Per-Client Training Time

C. Client Participation Matrix

D. Gradient Norm Evolution

E. Communication Cost Analysis

F. Learning Rate Sensitivity

G. Batch Size Impact

H. Per-Client Accuracy Trajectories

III. Extended Algorithm Comparison

A. Algorithms Evaluated

We compare foundational FL algorithms plus 2022–2025 advances:

Foundational: FedAvg [1], FedProx [2], SCAFFOLD [3], FedAdam/FedYogi/FedAdagrad [4].

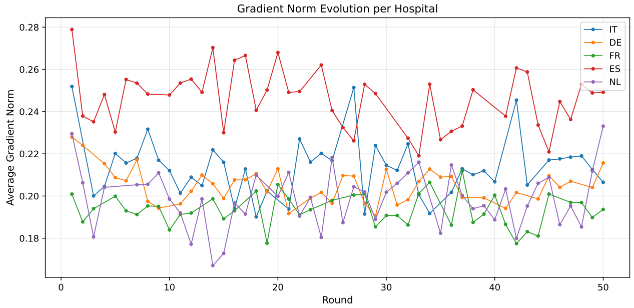

Fig. 4. Gradient norm evolution per client over 50 rounds. All clients show decreasing trends indicating stable convergence. Clipping threshold C=1.0 bounds extreme values for DP compatibility.

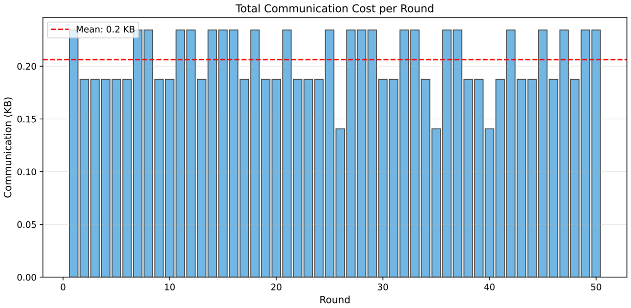

Fig. 5. Cumulative communication cost per round. Linear scaling with participating clients (3.5 KB/client/round). Total 50-round overhead: 875 KB for 5 clients—feasible even for bandwidth-constrained environments.

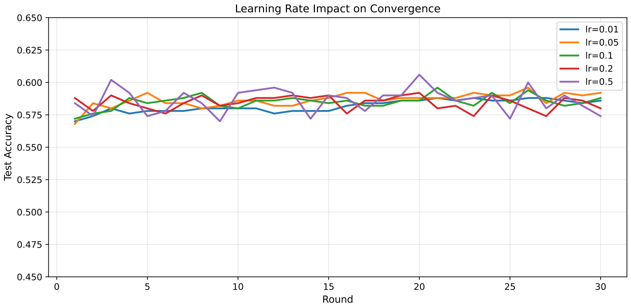

Fig. 6. Learning rate sensitivity analysis. η=0.01: slow convergence (53.8% at round 50). η=0.1: optimal (58.6%). η=0.5: instability with oscillations.

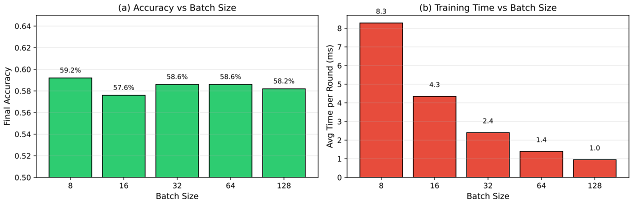

Fig. 7. Batch size impact on convergence. Smaller batches (8–16) provide noisier gradients but faster initial progress. Batch size 32 balances gradient quality and computational efficiency.

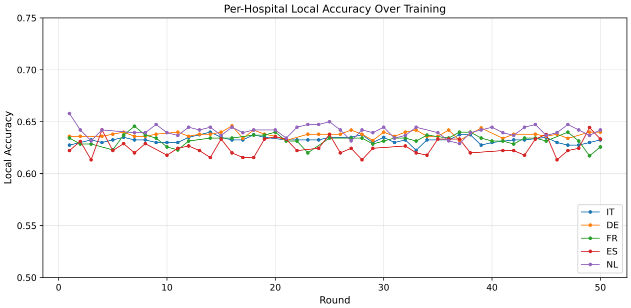

Fig. 8. Per-client accuracy over training rounds. Variance reflects non-IID data: NL (older, higher-risk population) reaches 64% accuracy while FR (mid-range demographics) stabilizes at 55%.

Recent (2022–2025): FedLC [9] (logit calibration for label skew), FedSAM [8] (flat minima), FedDecorr [10] (decorrelation against dimensional collapse), FedSpeed [11] (fewer rounds), FedExP [12] (server-side acceleration), FedLESAM [13] (globally-guided SAM, ICML 2024 Spotlight), HPFL [14] (personalized classifiers, ICLR 2025).

B. Non-IID Configuration

Data heterogeneity is controlled via Dirichlet distribution with α:

- $\alpha = 0.1$: **Extreme non-IID**—highly skewed label distributions
- $\alpha = 0.5$: **High non-IID**—significant heterogeneity
- $\alpha = 1.0$: **Moderate non-IID**—balanced heterogeneity
- $\alpha = 10.0$: **Near-IID**—approximately uniform

C. Convergence at Different Heterogeneity Levels

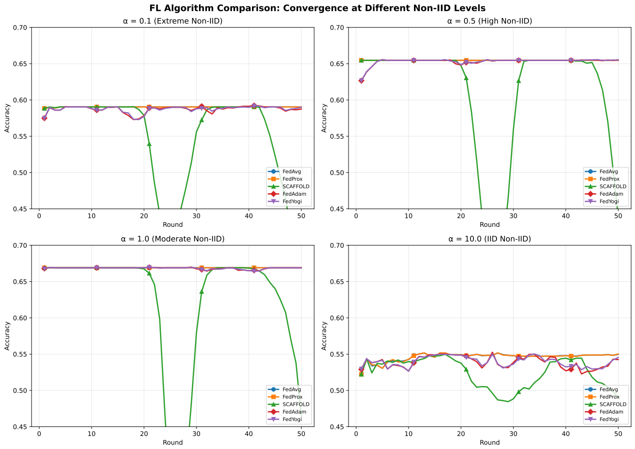

Fig. 9. Algorithm convergence across non-IID levels ($\alpha \in \{0.1, 0.5, 1.0, 10.0\}$). SCAFFOLD and adaptive methods show superior stability under extreme heterogeneity.

Findings: (1) At α=0.1, SCAFFOLD achieves most stable convergence via variance reduction. (2) FedProx provides marginal improvement over FedAvg at α=0.5–1.0. (3) Adaptive methods (FedAdam, FedYogi) excel in near-IID but may oscillate under extreme heterogeneity. (4) FedAvg remains competitive in near-IID, suitable for homogeneous federations.

D. Final Accuracy vs. Heterogeneity

E. Convergence Speed

F. Algorithm Selection Guidelines

Table I maps EHDS deployment scenarios to recommended algorithms.

IV. Advanced FL Paradigms

The reference implementation supports advanced FL paradigms beyond the 17 aggregation algorithms.

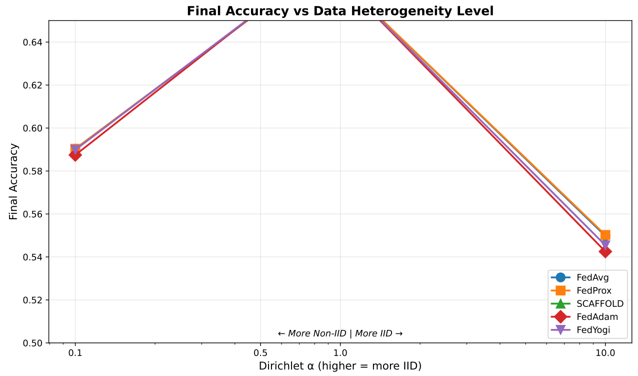

Fig. 10. Final accuracy vs. Dirichlet α. All algorithms degrade under extreme non-IID. SCAFFOLD shows smallest gap between α=0.1 and α=10.

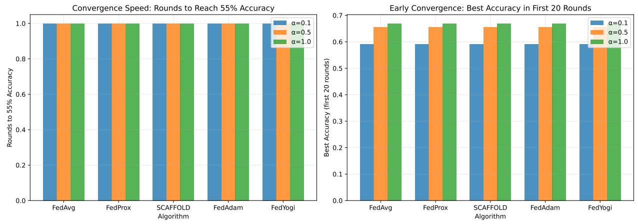

Fig. 11. Convergence speed comparison. Left: rounds to 55% accuracy. Right: best accuracy in first 20 rounds. Adaptive methods converge faster but may plateau.

A. Vertical Federated Learning

Vertical FL addresses scenarios where institutions hold complementary features for the same patients (e.g., hospital demographics + lab results + pharmacy records).

Private Set Intersection (PSI): RSA-based PSI identifies common patients without revealing full patient lists. Properties: neither party learns patients not in the intersection; $O(n \log n)$ complexity; pseudonymized identifiers for EHDS compliance.

TABLE I
ALGORITHM SELECTION FOR EHDS DEPLOYMENTS

EHDS Scenario	Algorithm	Rationale
Homogeneous MS	FedAvg	Simplicity, proven
Heterogeneous MS	SCAFFOLD	Variance reduction
Resource-limited	FedAdam	Fast convergence
Privacy-critical	FedAvg + DP	Well-studied bounds
Sparse participation	FedProx	Dropout resilience
Label-imbalanced	FedLC	Class-freq. calib.
Deep models, non-IID	FedDecorr	Dim. collapse prev.
Comm.-constrained	FedSpeed	Fewer rounds
No client changes	FedExP	Server-side only
SAM + global drift	FedLESAM	Global flatness
Per-hosp. classif.	HPFL	Local boundaries

MS = Member States. Scenarios may combine: heterogeneous + privacy-critical → SCAFFOLD + DP.

```
Algorithm S9: Split Learning Forward Pass
Input: Features $X_A$, $X_B$ at parties A, B; cut layer $k$
Output: Prediction $\hat{y}$
$h_A \leftarrow f^A_{1:k}(X_A)$        // Party A: features → activations
$h_B \leftarrow f^B_{1:k}(X_B)$        // Party B: features → activations
$h \leftarrow \mathrm{Concat}(h_A, h_B)$
$\hat{y} \leftarrow f_{k+1:L}(h)$        // Server: cut layer → output
return $\hat{y}$
```

B. Byzantine-Resilient Aggregation

Six defense methods protect against up to $f < n/3$ adversarial clients:

```
Algorithm S10: Krum Byzantine Defense
Input: Gradients $\{g_1, \ldots, g_n\}$, Byzantine bound $f$
Output: Selected gradient $g^*$
for each gradient $g_i$ do
    $D_i \leftarrow \{\|g_i - g_j\|^2 : j \neq i\}$
    $s_i \leftarrow \sum_{d \in \text{smallest}_{n-f-2}(D_i)} d$
$g^* \leftarrow g_{\arg\min_i s_i}$
return $g^*$
```

Other methods: **Trimmed Mean** (removes β-fraction extreme values per coordinate), **Coordinate-wise Median** (robust estimator), **Bulyan** (two-stage Krum + trimmed mean), **FLTrust** (server-guided trust weighting), **FLAME** (clustering-based). Attack simulation: label flipping, gradient scaling, additive noise, sign flipping, model replacement.

C. Continual Federated Learning

Healthcare data evolves over time—new diseases, changing protocols, demographic shifts. The EWC loss function adds a quadratic penalty:

$$\mathcal{L}_{EWC}(\theta) = \mathcal{L}(\theta) + \frac{\lambda}{2} \sum_i F_i(\theta_i - \theta_i^*)^2$$

where F_i is the Fisher Information for parameter i and θ^* are optimal parameters for previous tasks.

Additional strategies: Learning without Forgetting (LwF), Experience Replay, drift detection (ADWIN, Page-Hinkley) triggering adaptation.

D. Multi-Task Federated Learning

Different hospitals may target different prediction objectives. Architectures: **Hard Parameter Sharing** (common feature extractor, task-specific heads), **Soft Parameter Sharing** (separate networks with similarity regularization), **FedMTL** (dynamic task relationship learning).

E. Hierarchical Federated Learning

Four-tier hierarchy reflecting EU governance:

1) **Client Tier**: Individual hospitals/data holders
2) **Regional Tier**: Regional aggregators (e.g., Lombardy, Bavaria)
3) **National Tier**: National HDABs coordinate Member State aggregation

4) **EU Tier**: HealthData@EU central aggregator

Benefits: reduced communication costs (hospitals → regional, not directly EU), alignment with EHDS governance where HDABs have national jurisdiction.

F. Personalized Federated Learning

```
Algorithm S11: pFedMe Local Update
Input: Data $\mathcal{D}_k$, global $\theta$, personal $\theta_k$, $\lambda$, $\eta$
Output: Updated personal model $\theta'_k$
for $i = 1$ to $R$ do
    $\theta_k \leftarrow \theta_k - \eta\nabla\mathcal{L}(\theta_k; \mathcal{D}_k)$
// Moreau envelope: balance with global
$\theta'_k \leftarrow \theta_k - \lambda(\theta_k - \theta)$
$g_k \leftarrow \lambda(\theta - \theta'_k)$
return $\theta'_k$, $g_k$
```

Other approaches: **FedPer** (shared base, local personalization layers), **Per-FedAvg** (MAML-based meta-learning), **APFL** (adaptive mixing α between global and local), **Ditto** (personalization regularization).

EHDS Relevance: Member States have different healthcare systems, disease prevalence, and clinical practices. Personalized FL enables institution-specific adaptation while benefiting from collaborative training.

G. Asynchronous Federated Learning

```
Algorithm S12: FedAsync with Staleness Weighting
Input: Client update $\Delta_k$, client round $t_k$, server round $t$
Output: Updated global model $\theta$
$\tau \leftarrow t - t_k$                          // Staleness
$\alpha \leftarrow (1 + \tau)^{-a}$                // Polynomial decay, $a > 0$
$\theta \leftarrow \theta + \alpha \cdot \eta \cdot \Delta_k$
return $\theta$
```

Staleness functions: Constant (α=1), Polynomial ($(1+\tau)^{-a}$), Exponential ($e^{-a\tau}$), Hinge (1 if $\tau \leq \tau_{max}$, else 0). Additional: FedBuff (buffered async), semi-async (wait for α-fraction of clients).

H. Fairness-Aware Federated Learning

```
Algorithm S13: q-FedAvg Fair Aggregation
Input: Losses $\{L_1, \ldots, L_K\}$, updates $\{\Delta_1, \ldots, \Delta_K\}$, $q$
Output: Fair aggregated update $\Delta$
for each client $k$ do
    $w_k \leftarrow L_k^q$                 // Higher loss → higher weight
$W \leftarrow \sum_k w_k$; $w_k \leftarrow w_k / W$
$\Delta \leftarrow \sum_k w_k \cdot \Delta_k$
return $\Delta$
```

Fairness metrics: Performance Variance (Var($\{L_k\}$)), Worst-case Loss ($\max_k L_k$), Demographic Parity Gap, Equalized Odds Gap. Additional methods: AFL, FedMGDA+, TERM, FairFed.

Communication Manager Configuration

```
transport: gRPC | WebSocket
compression: gzip | lz4 | zstd | none
chunk_size: 1MB
retry_policy:
  max_retries: 3
  backoff: exponential
  base_delay: 1s
connection_pool:
  max_connections: 100
  idle_timeout: 300s
```

V. Infrastructure Components

A. Communication Layer

gRPC: Bidirectional streaming, Protocol Buffers (30% bandwidth reduction vs. JSON), HTTP/2 multiplexing. Ideal for data center deployments.

WebSocket: Browser-compatible, firewall-friendly (standard HTTP upgrade), event-driven. Ideal for edge deployments and browser-based participation.

B. Serialization

Binary Format: Tensor metadata + raw binary, 30% smaller than JSON, 15% smaller than pickle, cross-platform (Python, C++, Java).

Delta Serialization: Transmits only changed parameters, sparse encoding, up to 90% bandwidth reduction for fine-tuning.

EHDS-Compliant: Embeds permit ID, timestamp, provenance; cryptographic signatures; GDPR Article 30 audit fields.

C. Caching Layer

```
Algorithm S14: Distributed Lock for Aggregation
Input: Lock name, TTL, client ID
Output: Lock acquired (boolean)
acquired ← Redis.SET(lock_name, client_id, NX, EX=TTL)
if acquired then
    PerformAggregation()
    if Redis.GET(lock_name) == client_id then
        Redis.DEL(lock_name)
return acquired
```

Redis-based caching: model checkpoints, client states, real-time metrics. Features: LRU/LFU/TTL eviction, distributed locking, automatic serialization, cache warming.

D. Orchestration

Kubernetes: Deploys FL clients/aggregators as pods, HPA for elastic scaling, ConfigMaps for hyperparameters, Secrets for HDAB API keys.

Ray: Actor-based FL, automatic fault tolerance, Ray Tune for federated HPO, Object Store for gradient sharing.

Auto-Scaling: Reactive (queue depth/latency), Predictive (ML-based forecasting), Scheduled (time-based patterns).

E. Monitoring

Prometheus Metrics: Counters (rounds_total, permits_validated), Gauges (active_clients, privacy_budget_remaining), Histograms (round_duration, communication_latency), Summaries (gradient_norm_quantiles).

Grafana Dashboards: FL training progress, client health, latency heatmaps, privacy budget consumption, EHDS compliance status.

Alerting: Privacy budget exhaustion, client dropout threshold, model divergence, permit expiration.

F. Model Watermarking

IP protection for FL models trained on EHDS data: **Spread Spectrum** (frequency domain, robust to fine-tuning), **LSB** (low-order weight bits), **Backdoor-based** (input-output ownership proof), **Passport Layers** (dedicated ownership encoding).

G. Cross-Silo Enhancements

Multi-Model Federation: Weighted voting, stacking, mixture of experts with diversity enforcement.

Automatic Algorithm Selection: Data-driven selection via multi-armed bandit (UCB/Thompson Sampling), optimizing for accuracy, convergence, fairness, or privacy.

Algorithm S15: Adaptive Aggregation

Input: Client updates, metrics history, cooldown
Output: Aggregated model, selected algorithm

```
score ← WeightedScore(loss, accuracy, variance, conv.)
if RoundsSinceSwitch > Cooldown then
    for each candidate ∈ Algorithms do
        alt ← EstimatePerformance(candidate)
        if alt > score + Threshold then
            SwitchTo(candidate)
aggregated ← CurrentAlgo.Aggregate(updates)
return aggregated
```

VI. Extended EHDS Interoperability

A. OMOP Common Data Model

OMOP CDM v5.4 provides standardized analytical format used by European research networks (EHDEN, OHDSI).

ETL Pipelines: Transform source EHR to OMOP. **Vocabulary Mapping**: SNOMED, ICD10, LOINC, RxNorm. **Cohort Definitions**: ATLAS-compatible SQL generation. **Feature Extraction**: FeatureExtraction package for ML-ready datasets.

FL Integration: (1) Each hospital transforms local EHR to OMOP; (2) Feature extraction produces identical schema; (3) FL training on homogeneous feature spaces.

B. IHE Integration Profiles

ATNA: TLS mutual authentication, syslog audit messages (RFC 5424), maps to GDPR Article 30.

BPPC: Maps Article 71 opt-out to consent documents, XDS.b integration, consent enforcement at FL initiation.

XCA: Cross-border document query/retrieve, Initiating/Responding Gateways, patient identity correlation.

PIX/PDQ: Patient matching across boundaries, pseudonymization-aware identity management, national eHealth integration.

XUA: SAML 2.0 federated authentication, role-based access control, HDAB authorization token propagation.

C. Cross-Border Data Exchange

Message Formats: EHDS Data Permit Exchange Format (JSON-LD), Federated Query Protocol (SPARQL Federation), Model Update Message Format (Protocol Buffers).

Security: eIDAS-compliant electronic signatures, TLS 1.3, certificate-based authentication (EU trust framework).

Metadata: DCAT-AP Health extension, W3C PROV-O provenance, EMA data quality indicators.

D. Interoperability Architecture

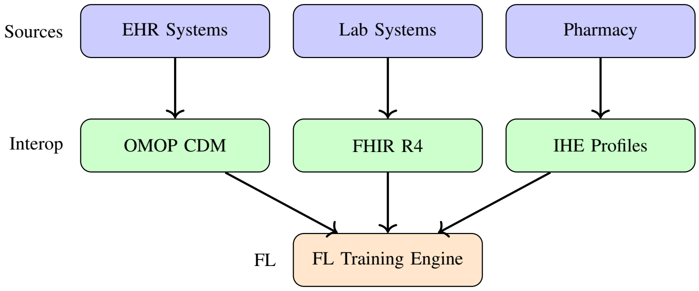

Fig. 12. Interoperability layer integrating heterogeneous data sources for FL training.

VII. Clinical Imaging: Extended Details

A. Datasets

Three clinical imaging datasets cover representative EHDS scenarios:

- **Chest X-ray** [17]: 5,860 pediatric radiographs (NORMAL/PNEUMONIA, 2.7:1 imbalance)
- **Brain Tumor MRI**: 3,064 T1-weighted CE MRI slices (3-class: glioma, meningioma, pituitary)
- **Skin Cancer**: 3,297 dermoscopy images (binary benign/malignant)

B. Model Architectures

HealthcareResNet: ResNet-18 [16] pretrained on ImageNet, GroupNorm replacing BatchNorm for FL stability. FedBN [15] skips normalization during aggregation. Partial backbone freeze (level 1). $\sim$11.2M parameters.

HealthcareCNN: 5-block CNN with GroupNorm, progressive channels (32→512), graduated Dropout (0.15→0.3). Classifier: Flatten→FC(512)→FC(128)→FC(K). $\sim$12M parameters.

Data augmentation: random horizontal flip, rotation (±15°), brightness jitter (±10%). ImageNet normalization.

C. V2 Experimental Configuration

- 5 hospitals, 25 rounds, 3 local epochs, batch size 32
- Adam optimizer (lr=0.001), early stopping (patience=6)
- Non-IID via Dirichlet α=0.5
- FedBN enabled, partial backbone freeze (level 1)
- 7 algorithms: FedAvg, FedProx, Ditto, FedLC, FedExP, FedLESAM, HPFL
- 3 seeds per configuration (42, 123, 456)
- Total: 7 × 5 datasets × 3 seeds = 105 experiments

D. Reproducibility

All experiments are fully reproducible:

```
cd fl-ehds-framework
# Full experiments (7 algo x 5 datasets x 3 seeds)
python -m benchmarks.run_full_experiments
# Quick validation (˜1-2h)
python -m benchmarks.run_full_experiments --quick
# Resume after interruption
python -m benchmarks.run_full_experiments --resume
```

Results, checkpoints, and logs are auto-saved to `benchmarks/paper_results/`.

Repository: https://github.com/FabioLiberti/FL-EHDS-FLICS2026

References

[1] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, pp. 1273–1282, 2017.

[2] T. Li *et al.*, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, vol. 2, pp. 429–450, 2020.

[3] S. P. Karimireddy *et al.*, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. ICML*, pp. 5132–5143, 2020.

[4] S. Reddi *et al.*, "Adaptive federated optimization," in *Proc. ICLR*, 2021.

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. NeurIPS*, vol. 32, pp. 14774–14784, 2019.

[6] I. Mironov, "Rényi differential privacy," in *Proc. IEEE CSF*, pp. 263–275, 2017.

[7] R. Hussein *et al.*, "Interoperability framework of the EHDS for secondary use," *J. Med. Internet Res.*, vol. 27, art. e69813, 2025.

[8] Z. Qu *et al.*, "Generalized federated learning via sharpness aware minimization," in *Proc. ICML*, PMLR 162, pp. 18250–18280, 2022.

[9] J. Zhang *et al.*, "Federated learning with label distribution skew via logits calibration," in *Proc. ICML*, PMLR 162, pp. 26311–26329, 2022.

[10] Y. Shi *et al.*, "Towards understanding and mitigating dimensional collapse in heterogeneous federated learning," in *Proc. ICLR*, 2023.

[11] Y. Sun *et al.*, "FedSpeed: Larger local interval, less communication round, and higher generalization accuracy," in *Proc. ICLR*, 2023.

[12] D. Jhunjhunwala, S. Wang, and G. Joshi, "FedExP: Speeding up federated averaging via extrapolation," in *Proc. ICLR*, 2023.

[13] Z. Qu *et al.*, "FedLESAM: Federated learning with locally estimated sharpness-aware minimization," in *Proc. ICML*, PMLR 235, 2024.

[14] Y. Chen, X. Cao, and L. Sun, "HPFL: Hot-pluggable federated learning with shared backbone and personalized classifiers," in *Proc. ICLR*, 2025.

[15] X. Li *et al.*, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. ICLR*, 2021.

[16] K. He *et al.*, "Deep residual learning for image recognition," in *Proc. CVPR*, pp. 770–778, 2016.

[17] D. Kermany *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.