Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077337

ARTICLE

Federated Learning for Malicious Domain Detection via Privacy-Preserving DNS Traffic Analysis

Samar Abbas Mangi[1,*], **Samina Rajper**[1], **Noor Ahmed Shaikh**[1] **and Shehzad Ashraf Chaudhry**[2,3]

[1]Institute of Computer Science, Shah Abdul Latif University Khairpur, Khairpur, Pakistan
[2]Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[3]Department of Software Engineering, Nisantasi University, Istanbul, Türkiye

*Corresponding Author: Samar Abbas Mangi. Email: mangisamar@gmail.com

Received: 07 December 2025; Accepted: 09 February 2026; Published: 09 April 2026

ABSTRACT: Malicious domain detection (MDD) from DNS telemetry enables early threat hunting but is constrained by privacy and data-sharing barriers across organizations. We present a deployable federated learning (FL) pipeline that trains a compact deep neural network (DNN; 64-32-16 with ReLU and dropout 0.3) locally at each client and exchanges only masked model updates. Privacy is enforced via secure aggregation (the server observes only an aggregate of masked updates) and optional server-side differential privacy (DP) via clipping and Gaussian noise. Our feature schema combines DNS-specific lexical cues (character *n*-grams, entropy, TLD indicators) with lightweight behavioral signals (TTL dispersion, query cadence) without exporting raw logs or identifiers. We benchmark FedAvg, FedProx, and FedNova under controlled non-IID client partitions and report ROC-AUC, precision-recall area under the curve (PR-AUC), F1, convergence speed, and communication cost. Federated models approach centralized training while outperforming local-only baselines; FedProx reaches the target Accuracy ≥ 0.995 in fewer rounds than FedAvg under medium heterogeneity. We report 95% bootstrap confidence intervals and paired significance tests (DeLong for ROC-AUC; McNemar for Accuracy). Overall, privacy-preserving FL for DNS-based MDD is practical, providing near-centralized utility while keeping DNS data local.

KEYWORDS: Federated learning; DNS security; malicious domain detection; privacy-preserving analytics; secure aggregation; differential privacy

1 Introduction

Domain Name System (DNS) telemetry is a first responder for network defense: almost every malicious campaign-from phishing and command & control (C2) to domain-generation algorithms (DGAs) and fast-flux infrastructure-ultimately surfaces as resolvers translating hostnames to IPs. Learning discriminative patterns from DNS logs can therefore identify malicious domains early, complementing endpoint and flow-based analytics. However, DNS data is privacy-sensitive (e.g., reveals user intent, enterprise asset names) and often siloed by policy, regulation, or business constraints. Centralizing raw logs across organizations is thus impractical or outright prohibited, leaving most detection models data-starved and biased to single-tenant environments.

Federated learning (FL) offers a pragmatic alternative: clients (enterprises, ISPs, campuses) train locally on their own DNS events and share only model updates with a coordinating server. In principle, FL preserves data locality while unlocking cross-organization generalization. In practice, however, several challenges

remain under-explored for the *DNS malicious domain detection* (MDD) setting: (i) **feature realism**, where lexical signals (e.g., character n-grams, entropy) must be combined with temporal/query-behavior cues (e.g., TTL dispersion, burstiness) without leaking raw queries; (ii) **non-IID heterogeneity**, as clients differ in user populations, resolvers, time-zones, and threat exposure, leading to skewed label/base-rate distributions and drift; (iii) **privacy–utility trade-offs**, since secure aggregation and differential privacy (DP) introduce noise or constraints that may hinder convergence; and (iv) **robustness**, where adversarial clients can poison model updates (label-flip, sign-flip, backdoors) unless the aggregation stack includes integrity checks or robust estimators.

Prior FL-for-security efforts (e.g., CO-DEFEND, FedMSE) demonstrate the promise of distributed training but typically evaluate on flow records or generic security telemetry, use one FL optimizer (FedAvg), and limit privacy/robustness analysis or DNS-specific features. Moreover, DNS-focused studies frequently evaluate centralized models on curated feeds, which overestimate generalization and ignore cross-tenant heterogeneity. Consequently, the community lacks a *deployable, privacy-preserving, and robustness-aware* FL pipeline tailored to DNS MDD with comprehensive baselines and statistical rigor.

1.1 Problem Statement

Let $\mathcal{C} = \{1, \ldots, N\}$ denote clients (organizations) with local datasets $\mathcal{D}_i$ of domain-level feature vectors $x \in \mathbb{R}^d$ and labels $y \in \{0, 1\}$ (benign/malicious). The goal is to learn a global classifier $f_\theta : \mathbb{R}^d \to \{0, 1\}$ that maximizes utility (AUC/F1) under constraints: (i) raw $\mathcal{D}_i$ never leaves client i; (ii) server only observes masked/aggregated updates; and (iii) aggregation is resilient to reasonable levels of client noise or malice. We specifically study lightweight models suitable for commodity client hardware and limited uplink bandwidth.

1.2 Threat Model and Privacy Scope

We assume an *honest-but-curious* server that executes the protocol but tries to infer client information from messages, and clients that can be benign or adversarial. We employ (a) *secure aggregation* so the server observes only the *sum* of masked updates, and (b) *DP at the server* via clipping and Gaussian noise to bound information leakage over rounds. We analyze poisoning pressure via simple label-flip/sign-flip stress tests and discuss robust aggregation options (median/trimmed-mean).

1.3 Design Principles

Our pipeline prioritizes: (1) *deployability* via a compact DNN (64–32–16 with ReLU and dropout 0.3) that fits typical enterprise clients; (2) *realism* via DNS-specific lexical and temporal features engineered to avoid raw-log exfiltration; (3) *coverage of FL baselines* (FedAvg, FedProx, FedNova) under controlled non-IID regimes; and (4) *statistical transparency* using 95% confidence intervals (CIs) and paired significance tests against a centralized reference.

1.4 Contributions

- **C1: DNS-specific FL pipeline.** We present an end-to-end, privacy-preserving pipeline for malicious domain detection that combines *lexical* (e.g., n-gram profiles, entropy, TLD indicators) and *temporal/behavioral* features (e.g., TTL spread, query cadence) within a deployable lightweight DNN appropriate for bandwidth- and compute-constrained clients.
- **C2: Comprehensive FL baselines under non-IID.** Beyond FedAvg, we evaluate *FedProx* (proximal regularization for client drift) and *FedNova* (normalization for heterogeneous local steps) across mild/medium/severe label and feature skew, reporting both accuracy metrics and *communication rounds/bytes* to a target AUC.

- **C3: Statistical rigor.** We report 95% bootstrap CIs and paired significance tests (DeLong for ROC-AUC; McNemar for accuracy) vs. centralized and local-only baselines, and we characterize per-client variance to expose heterogeneity effects.
- **C4: Privacy & robustness analysis.** We give a concrete, step-by-step *secure aggregation* example (masking and cancellation), apply *server-side DP* (clipping+Gaussian noise), and examine *poisoning awareness* via anomaly screening and robust aggregation discussion, quantifying utility impact where feasible.
- **C5: Reproducible artifacts.** We outline feature schemas, FL hyperparameters, and non-IID partition recipes to facilitate replication and extension in real-world pilots.

1.5 Research Questions (RQs)

RQ1: How close can privacy-preserving FL get to a centralized upper bound on DNS MDD while outperforming local-only training?

RQ2: Which FL optimizer (FedAvg, FedProx, FedNova) best tolerates DNS-style non-IID skew when factoring in communication cost?

RQ3: What is the empirical cost of secure aggregation and server-side DP on utility and convergence?

RQ4: Under simple poisoning stress (label/sign flip), which lightweight mitigations stabilize performance without heavy server overhead?

1.6 Organization

Section 2 positions our work against recent FL-for-security literature and DNS detection studies. Section 3 details datasets, feature design, client partitions, and non-IID generators. Section 4 describes the lightweight DNN and the FL training protocol (FedAvg/FedProx/FedNova). Section 5 formalizes secure aggregation and server-side DP and outlines poisoning-aware screening. Section 6 presents results with CIs/significance, ablations on model choice and non-IID severity, and a communication-cost audit. Section 7 discusses deployment, limitations, and ethical considerations; Section 8 concludes.

2 Related Work

Centralized DNS malicious domain detection. Malicious domain detection (MDD) using DNS telemetry has traditionally been studied in centralized settings where resolver logs are aggregated and models are trained using lexical statistics (e.g., length distributions, character n-grams, Shannon entropy), tokenized subdomain structure, and auxiliary context. Surveys summarize these feature families and pipelines, and also highlight limitations when models are deployed across populations with different base rates, naming conventions, and operational policies [1,2]. While centralized training can achieve strong in-tenant performance, regulatory, contractual, and data-governance constraints frequently prevent cross-organization pooling of DNS logs, motivating learning paradigms that preserve data locality.

Federated learning for security and intrusion detection. Federated learning (FL) addresses this constraint by exchanging model updates rather than raw data; FedAvg remains the canonical baseline for cross-silo deployments [3]. Empirical evidence from privacy-sensitive domains (e.g., cross-institutional healthcare) further supports the feasibility of training competitive models without sharing sensitive records [4]. Building on these foundations, a growing body of work adapts FL to intrusion detection and network/IoT security. Recent surveys and systematic reviews emphasize recurring challenges such as non-IID client distributions, participation heterogeneity, and communication overhead, and provide taxonomies of design choices and threat models [5–8]. Representative systems explore diverse datasets and architectures, including decentralized approaches for encrypted-DNS-related detection, representation

learning for IoT IDS, and federated deep architectures for wireless IDS, as well as coordination mechanisms (e.g., blockchain-backed auditing) aimed at improving provenance and integrity [9–13].

Threat-specific FL applications and privacy-aware DNS analytics. Beyond generic IDS, threat- and topology-specific studies broaden the landscape. Adaptive FL has been investigated for DDoS detection under workload variation [14], and hybrid deep models combined with FL have been evaluated in SDN contexts [15]. Privacy-preserving FL formulations have also been explored for content-layer threats such as phishing [16]. In parallel, encrypted DNS and privacy-aware DNS analytics underscore the tension between visibility and confidentiality, reinforcing the need for feature engineering and release constraints that avoid exposing raw resolver logs [17]. From a systems perspective, decentralized and 6G-oriented analyses further study how topology and transport interact with FL reliability, bandwidth budgets, and privacy [18].

Robustness and privacy enforcement in federated security. Robustness and privacy enforcement are active research directions in FL-based security. Reviews catalog defenses against malicious or faulty clients, including update screening and robust aggregation, and discuss how these mechanisms interact with client skew and operational constraints [19]. Complementary work investigates combining FL with differential privacy (DP) and related privacy mechanisms in security/IoT settings to improve confidentiality while maintaining detection performance [20]. Additional trends explore orthogonal directions such as privacy-enhanced botnet detection, hierarchical federated generative modeling in smart environments, and broader cyber-physical settings that benefit from privacy-preserving learning patterns [21–23]. Early position and working papers also discuss next-generation AI for DNS-over-HTTPS and forensics-oriented anomaly detection, reflecting continuing interest in secure and privacy-aware DNS analytics [24,25].

Gap and positioning. Despite steady progress on FL for IDS, comparatively fewer studies provide a DNS-specific, end-to-end FL pipeline that jointly (i) uses privacy-compatible DNS feature representations (to avoid raw-log sharing), (ii) evaluates multiple FL optimizers under controlled non-IID regimes with rigorous uncertainty reporting, and (iii) integrates secure aggregation and optional DP with explicit communication accounting while examining practical robustness stabilizers. Our work targets this gap by designing DNS-specific lexical and temporal features suitable for cross-silo deployment, systematically comparing FL optimizers under varying heterogeneity, and incorporating secure aggregation and optional server-side DP alongside lightweight screening and robust aggregation mechanisms.

3 Datasets and Federation Setup

This section describes how domains are sourced and labeled, how per–client DNS datasets are constructed under privacy constraints, and how federation parameters (client sampling, non–IID partitions, and training protocol) are instantiated.

3.1 Source Domains and Labeling Pipeline

Benign and malicious candidates are collected from multiple feeds and then normalized into a common schema to avoid raw–log sharing.

3.1.1 Benign Pool

Benign domains are drawn from popularity lists (e.g., top sites by category and geography) and curated enterprise allowlists. To reduce popularity bias and near–duplicate leakage, we (i) stratify by category (*news*, *commerce*, *cloud*, *social*) and region, (ii) deduplicate at the *registered domain* level using the public suffix list, and (iii) remove ephemeral tracking subdomains via rule–based filters.

3.1.2 Malicious Pool

Malicious candidates are aggregated from multiple threat–intelligence sources, with emphasis on DGAs and fast–flux infrastructure. To cover families unseen in a single feed, we (i) merge across feeds and time windows, (ii) include synthetically generated DGA samples using public DGA seeds (placed *only* in training unless otherwise noted), and (iii) apply retrospective labels for domains later confirmed by incident reports. Conflicts are resolved with a precedence order `malicious > benign > unknown`, with `unknown` subsequently discarded.

3.1.3 Label Assignment

Each domain d receives a binary label $y(d) \in \{0,1\}$ via a two–stage rule: (i) *feed membership* (hard label if present in a high–confidence source), and (ii) *lightweight heuristics* (lexical entropy, suspicious n–gram frequency, newly observed TLDs) used only to prioritize manual audit, not as ground truth. Final labels are stored as (d, y, t) where t is the label timestamp. We keep per–source provenance to support ablations. The overall per-client dataset construction workflow is summarized in Fig. 1.

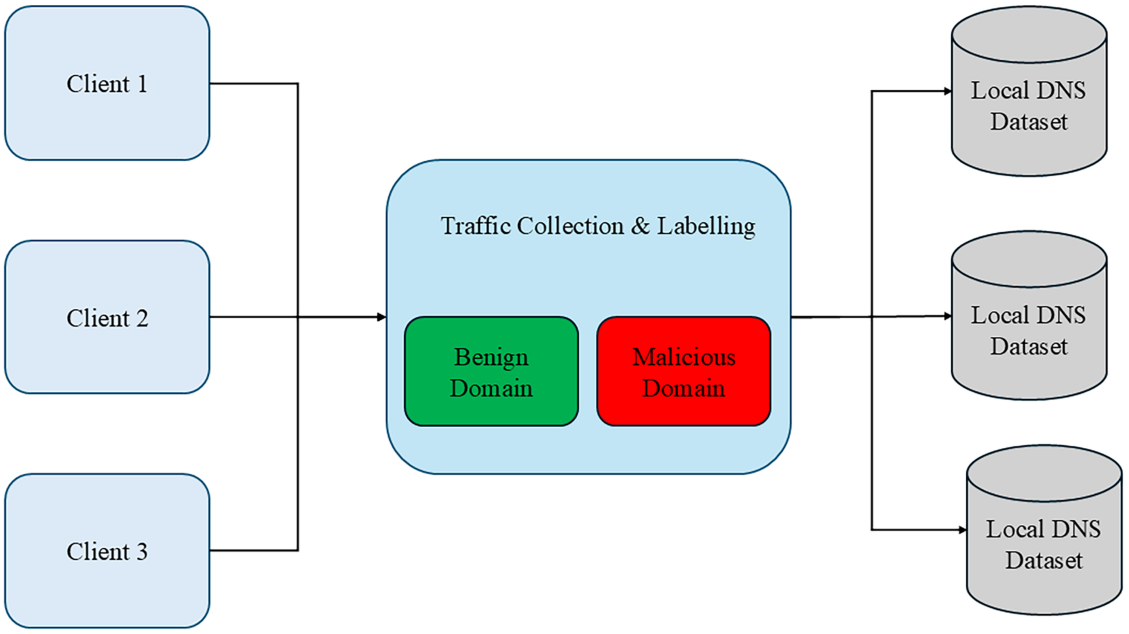

Figure 1: Per–client DNS dataset creation from a central labeling pipeline. Centralized labeling is performed on feed identifiers and privacy–compatible heuristics; only feature vectors and labels are retained per client.

3.2 Feature Schema (Privacy–Compatible)

For each domain we compute features locally from DNS resolver summaries to avoid raw queries or PII. Let $\phi(d) \in \mathbb{R}^{d_\phi}$ denote the feature vector.

- *Lexical* (computed on the registered domain and subdomain tokens): length, vowel/consonant ratios, Shannon entropy, n–gram frequencies ($n \in \{2,3,4\}$), digit fraction, hyphen count, TLD one–hot/embedding, character class run–lengths.
- *Behavioral/temporal* (aggregated over a window W=24 h): TTL mean/variance, unique resolver count, query cadence statistics (inter–arrival mean/variance, burstiness index), response NXDomain ratio, first–seen recency, diurnal skew.

- *Infrastructure hints* (if available without leaking PII): AS number histogram (coarsened), IP geodiversity counts (bucketized), passive DNS co–occurrence degree (k–anonymized).

Client-side overhead. Behavioral features are computed from window-level counters (streaming aggregates) rather than per-packet processing; the dominant cost is maintaining per-registered-domain counts within each window. This scales with the number of distinct domains observed in the window and is practical for high-throughput resolvers when implemented as incremental counters (e.g., hash maps with periodic compaction).

All features are standardized per client using training–fold statistics only. Fields that could reveal users or hostnames are *never* exported.

3.3 Train/Validation/Test Protocol

Datasets are split temporally to reduce leakage from future into past. For each client i:

$$\mathcal{D}_i^{\text{train}} = \{(d, \phi(d), y, t) \mid t \in [T_0, T_1)\}, \tag{1}$$

$$\mathcal{D}_i^{\text{val}} = \{(d, \phi(d), y, t) \mid t \in [T_1, T_2)\}, \tag{2}$$

$$\mathcal{D}_i^{\text{test}} = \{(d, \phi(d), y, t) \mid t \in [T_2, T_3)\}. \tag{3}$$

A representative choice is T_0 = `2021-01-01`, T_1 = `2021-07-01`, T_2 = `2022-01-01`, T_3 = `2022-07-01`. Class imbalance is handled with focal loss or cost–sensitive weighting on $\mathcal{D}_i^{\text{train}}$; we report threshold–free metrics (ROC–AUC, PR–AUC) on $\mathcal{D}_i^{\text{test}}$.

3.4 Non–IID Client Partitions

To emulate realistic heterogeneity across organizations, we generate per–client label/feature skews by sampling client–specific class mixtures and temporal slices. A common construction uses a Dirichlet prior:

$$\boldsymbol{\pi}_i \sim \text{Dir}(\alpha \mathbf{1}_K), \qquad \mathbb{P}(y{=}k \mid i) = \pi_{i,k}, \tag{4}$$

where K is the number of classes (here, $K = 2$) and α controls skew (smaller $\alpha \Rightarrow$ stronger non–IID). We evaluate *mild*, *medium*, and *severe* non–IID with $\alpha \in \{10.0, 1.0, 0.3\}$. Feature–level skew is induced by stratifying on TLDs, DGA families, and time zones. Client dataset sizes $|\mathcal{D}_i|$ follow a log–normal distribution to reflect real deployment variance.

3.5 Client Privacy, Sanitization, and Governance

Before feature extraction, resolvers apply on–premises sanitization: strip query names to registered domains, hash auxiliary fields with keyed hashing where needed, and aggregate per window $W = 24$ h. The exported artifact never contains IPs, user identifiers, or raw QNAMEs. Provenance and label timestamps are retained locally and summarized in privacy–compatible counters for evaluation.

3.6 Federation Topology and Participation

We adopt a standard cross–silo FL topology with a central aggregator and N long–lived clients. At each round r the server samples a subset $\mathcal{S}_r$ with participation rate q:

$$\mathcal{S}_r \subseteq \{1, \ldots, N\}, \qquad |\mathcal{S}_r| = \max\left(1, \lfloor qN \rfloor\right), \tag{5}$$

and broadcasts the current global weights $\theta^{(r)}$. Each participating client runs E local epochs with batch size B and learning rate η, producing an update $\Delta\theta_i^{(r)}$.

3.7 Secure Aggregation and Differential Privacy

Clients mask their updates using pairwise or PRG–derived masks so that the server can only compute the sum:

$$\tilde{\Delta\theta}_i^{(r)} = \Delta\theta_i^{(r)} + m_i^{(r)}, \qquad \sum_{i\in\mathcal{S}_r} m_i^{(r)} = 0. \tag{6}$$

The aggregator computes $\sum_{i\in\mathcal{S}_r} \tilde{\Delta\theta}_i^{(r)}$ without access to any individual update. We apply server–side DP by clipping the aggregated update to $\|\cdot\|_2 \le C$ and adding Gaussian noise $\mathcal{N}(0, \sigma^2 I)$:

$$\widehat{\Delta\theta}^{(r)} = \mathrm{clip}\Big(\sum_{i\in\mathcal{S}_r} \tilde{\Delta\theta}_i^{(r)}, C\Big) + \mathcal{N}(0, \sigma^2 I), \tag{7}$$

with privacy accounting over R rounds reported as $(\varepsilon, \delta) = (3.5, 10^{-5})$ for sampling rate $q = 1.0$, clipping $C = 1.0$, and noise scale $\sigma = 0.5$.

3.8 Optimizers and Hyperparameters

We compare three FL variants:

- *FedAvg*: standard weighted averaging of client updates.
- *FedProx*: proximal regularization with coefficient $\mu = 10^{-3}$ to stabilize heterogeneous objectives.
- *FedNova*: normalization to correct bias from variable local steps.

Unless noted, we use $E = 1$, $B = 680$, $\eta = 10^{-3}$, $q = 1.0$, and $R = 10$ rounds. The local model is a compact DNN (64–32–16 with ReLU and dropout 0.3), chosen for deployability on commodity client hardware.

3.9 Cross–Time Generalization and Leakage Controls

To evaluate robustness to temporal shift, we hold out a future slice for testing and prevent leakage by (i) forbidding duplicated registered domains across splits, (ii) excluding synthetic DGA samples from test unless explicitly stated, and (iii) recomputing feature standardization statistics *only* on the training fold. Model selection uses validation AUC and early stopping with patience 3 rounds.

Table 1 summarizes the dataset composition used in our experiments (global totals and per-client statistics) for reproducibility.

Table 1: Dataset summary for reproducibility (global totals and per-client statistics).

Item	Benign	Malicious
Total domains (registered)	**1,000,000**	**1,000,000**
Train split (domains)	699,658	699,730
Validation split (domains)	150,112	150,016
Test split (domains)	150,230	150,254
Per-client dataset size (domains): min 199,369, mean 200,000.00, max 200,860		
Number of clients: $N = 10$ (simulated cross-silo)		

Note: Counts are reported at the registered-domain level after deduplication and temporal splitting.

3.10 Simulation Realism and Bias

Synthetic query generation and replay can under–approximate resolver–specific caching, TTL handling, and burstiness. We therefore (i) evaluate cross–time generalization by training on $[T_0, T_2)$ and testing on

$[T_2, T_3)$, (ii) include, when permissible, a small passive–capture subset of size 10^5 domains to sanity–check trends, and (iii) report sensitivity to class priors by reweighting the test distribution to client–specific base rates. Limitations and ethical considerations are discussed in Section 7.

4 Model and Training

This section specifies the lightweight classifier architecture, the supervised learning objective and regularization, the calibration and thresholding strategy for deployment, and the federated optimization protocols (FedAvg, FedProx, FedNova). We also describe client participation, early stopping, and communication/computation accounting.

4.1 Classifier Architecture

The local model is a compact feed-forward DNN designed for modest client hardware. Let $x \in \mathbb{R}^{d_\phi}$ denote the feature vector from Section 3. The network computes

$$h_1 = \mathrm{ReLU}(W_1 x + b_1), \qquad h_1 \leftarrow \mathrm{Dropout}(h_1;\, p = 0.3), \tag{8}$$

$$h_2 = \mathrm{ReLU}(W_2 h_1 + b_2), \qquad h_2 \leftarrow \mathrm{Dropout}(h_2;\, p = 0.3), \tag{9}$$

$$h_3 = \mathrm{ReLU}(W_3 h_2 + b_3), \tag{10}$$

$$z = W_4 h_3 + b_4, \qquad \hat{y} = \sigma(z), \tag{11}$$

with layer widths $(d_\phi \to 64 \to 32 \to 16 \to 1)$. The sigmoid $\sigma(\cdot)$ outputs the malicious probability. We initialize weights with He/Kaiming initialization and use layer-norm optionally (disabled by default for latency). The architecture is depicted in Fig. 2.

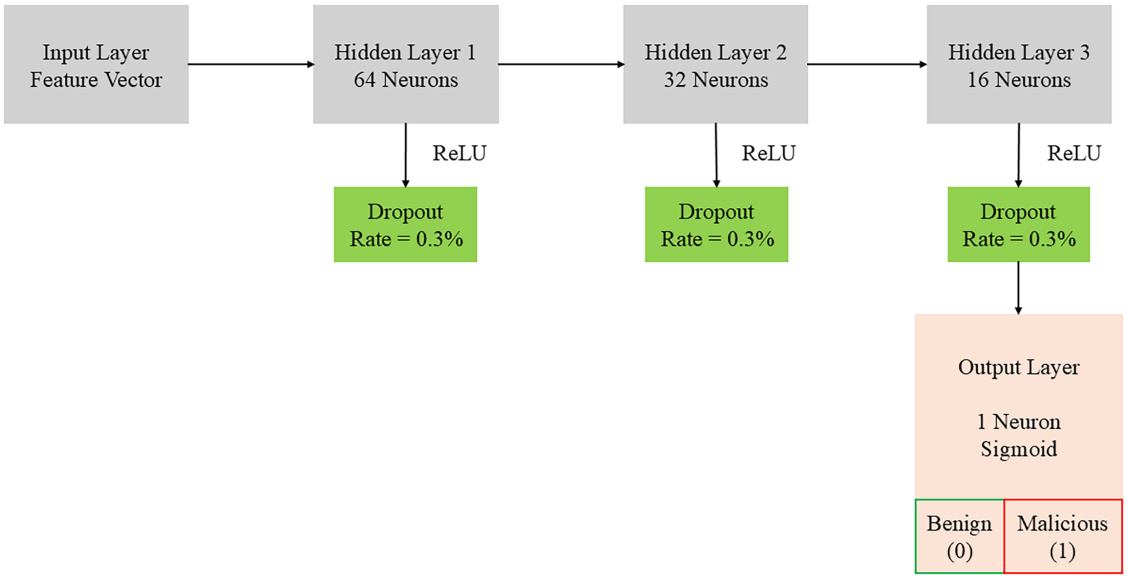

Figure 2: Lightweight DNN used for binary domain classification (benign vs. malicious).

4.2 Objective, Class Imbalance, and Regularization

Binary cross-entropy with class weights addresses imbalance:

$$\mathcal{L}_{\mathrm{BCE}}(\theta) = -\frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \Big(w_1\, y\, \log(\hat{y}) + w_0\, (1-y)\, \log(1-\hat{y}) \Big), \tag{12}$$

where $\hat{y} = \sigma(z)$ is the predicted malicious probability and (w_1, w_0) are inverse-frequency weights computed *per client* on the training fold. As a drop-in alternative, focal loss $\mathcal{L}_{\mathrm{focal}}$ with focusing parameter $\gamma = 2.0$ is supported in ablations. ℓ_2 weight decay is applied to all layers with coefficient $\lambda = 10^{-4}$.

4.3 Optimization and Scheduling (Local)

Each client trains with mini-batch SGD or Adam. Default settings are Adam ($\eta = 10^{-3}$, $\beta_1 = 0.9, \beta_2 = 0.999$), batch size $B = 680$, local epochs per round $E = 1$. A cosine or step scheduler can be enabled to reduce η across rounds; by default we keep η fixed for stable aggregation.

4.4 Calibration and Thresholding

For deployment, we calibrate probabilities on the validation set using temperature scaling: $z' = z/T$, $T > 0$ selected by minimizing NLL on $\mathcal{D}^{\mathrm{val}}$. The decision threshold τ is set to maximize F1 or to satisfy a target recall $r_\star = 0.98$ (if operational policy prioritizes coverage). Unless stated, metrics are threshold-free (ROC-AUC, PR-AUC) on test.

4.5 Federated Optimization

Let $\theta^{(r)}$ denote global parameters at round r. The server samples a subset $\mathcal{S}_r$ of clients with participation rate q and broadcasts $\theta^{(r)}$. Each participating client $i \in \mathcal{S}_r$ performs E local epochs on its dataset $\mathcal{D}_i$ and returns an update or local solution.

4.5.1 FedAvg

Clients compute local minimizers (or E-epoch solutions) $\theta_i^{(r+1)}$; the server aggregates by client-size weights $n_i = |\mathcal{D}_i|$:

$$\theta^{(r+1)} = \sum_{i\in\mathcal{S}_r} \frac{n_i}{\sum_{j\in\mathcal{S}_r} n_j}\, \theta_i^{(r+1)}. \tag{13}$$

4.5.2 FedProx

Clients solve a proximal subproblem that stabilizes training under heterogeneous objectives and non-IID data:

$$\min_{\theta}\; F_i(\theta) \;+\; \frac{\mu}{2}\, \|\theta - \theta^{(r)}\|_2^2, \tag{14}$$

where F_i is the empirical risk on $\mathcal{D}_i$ and $\mu = 10^{-3}$ is the proximal coefficient. Aggregation follows FedAvg weighting.

4.5.3 FedNova

To correct bias from variable local steps and partial participation, FedNova normalizes client contributions by their effective local step counts a_i:

$$\Delta_i = \frac{\theta_i^{(r+1)} - \theta^{(r)}}{a_i}, \qquad \theta^{(r+1)} = \theta^{(r)} + \sum_{i \in \mathcal{S}_r} \omega_i \Delta_i, \tag{15}$$

with ω_i typically proportional to n_i. We measure convergence speed and communication to a target AUC of 0.9998 for each optimizer.

4.6 Client Sampling, Early Stopping, and Rounds

At round r, the server samples $\mathcal{S}_r$ uniformly without replacement with $|\mathcal{S}_r| = \max(1, \lfloor qN \rfloor)$ and $q = 1.0$ in our default experiments. Training proceeds for $R = 10$ rounds or until validation AUC plateaus on a moving window of 3 rounds (patience). We report the best-val checkpoint per method and re-score on the test slice.

4.7 Ablations: Architecture and Loss

Ablations compare the DNN against (i) a 1D-CNN over character-level embeddings of the domain string and (ii) a GRU sequence model over tokenized subdomains. For each model we record parameter count, client-side latency (ms) on a reference CPU (Intel Xeon Silver), ROC-AUC/PR-AUC on the test split, and rounds to a target AUC 0.9998.

4.8 Communication and Computation Accounting

Let P be the number of parameters. Per round, each selected client uploads P floating-point values (or a compressed/quantized representation if enabled) and downloads P values:

$$\text{Uplink}_{\text{MB/round}} \approx \frac{|\mathcal{S}_r| \cdot P \cdot b}{8 \cdot 10^6}, \quad \text{Downlink}_{\text{MB/round}} \approx \frac{P \cdot b}{8 \cdot 10^6}, \tag{16}$$

where b is bits per parameter (32 by default, 8 if int8 quantization is used for payloads). Client compute per round is E epochs over $|\mathcal{D}_i|$ with complexity $\mathcal{O}(E\,|\mathcal{D}_i|\,P)$ for dense layers. We report total uplink/downlink to reach the target AUC and the wall-clock under participation q.

4.9 Implementation Details and Defaults

Unless stated, training uses Adam with learning rate $\eta = 10^{-3}$, batch size $B = 680$, local epochs $E = 1$, participation $q = 1.0$, rounds $R = 10$, weight decay $\lambda = 10^{-4}$, and dropout $p = 0.3$. Mixed precision is disabled by default for reproducibility; gradient clipping at norm $c_g = 1.0$ is enabled to stabilize local training under rare spikes. Unless otherwise noted, clients train on CPU; GPU results are reported separately where applicable. The end-to-end federated learning workflow is illustrated in Fig. 3.

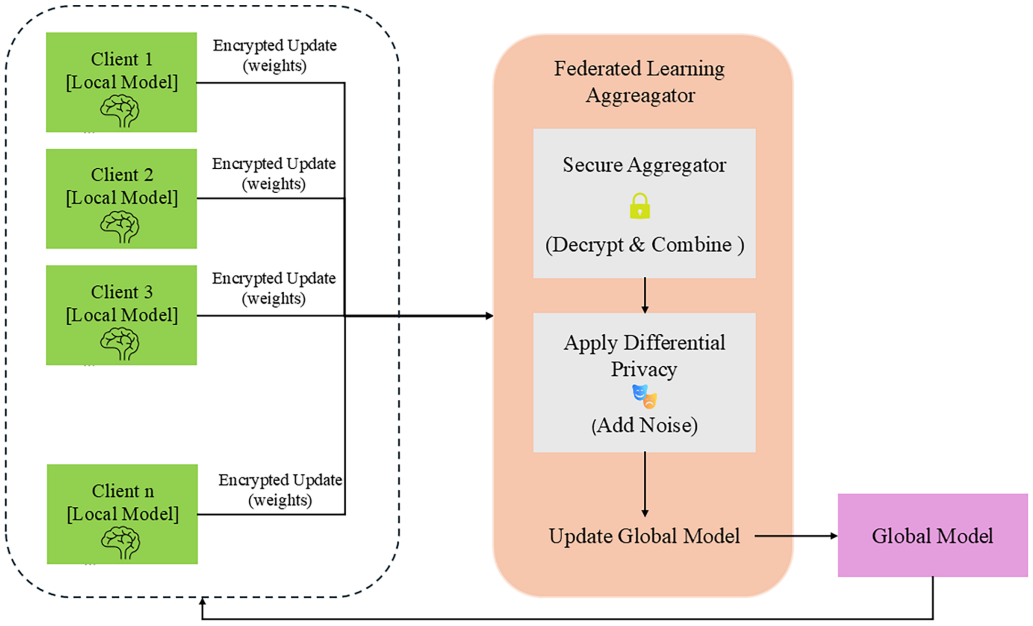

Figure 3: High-level FL system overview with broadcast.

5 Privacy and Robustness Mechanisms

5.1 Secure Aggregation

Each client masks its local update so that the server can only recover an aggregate over participating clients. Client i computes an update $\Delta\theta_i^{(r)}$ at round r and transmits a masked update

$$\widetilde{\Delta\theta}_i^{(r)} = \Delta\theta_i^{(r)} + m_i^{(r)}, \qquad \text{where} \qquad \sum_{i\in\mathcal{S}_r} m_i^{(r)} = 0. \tag{17}$$

The server aggregates $\sum_{i\in\mathcal{S}_r} \widetilde{\Delta\theta}_i^{(r)}$, which equals $\sum_{i\in\mathcal{S}_r} \Delta\theta_i^{(r)}$ because the masks cancel. In operational cross-silo settings, dropout-resilient secure aggregation variants are commonly used so aggregation remains correct even if a subset of clients fails to complete a round; our experiments model partial participation and churn to reflect this behavior. The privacy-preserving aggregation process is illustrated in Fig. 4.

5.1.1 Worked Example (Explicit Numeric)

Consider three clients with 2D updates $\Delta\theta_1 = (1.0, -0.5)$, $\Delta\theta_2 = (0.2, 0.1)$, and $\Delta\theta_3 = (-0.7, 0.4)$. Let masks be $m_1 = (0.3, -0.2)$, $m_2 = (-0.6, 0.5)$, $m_3 = (0.3, -0.3)$ so that $m_1 + m_2 + m_3 = (0, 0)$. Clients send masked updates: $\widetilde{\Delta\theta}_1 = (1.3, -0.7)$, $\widetilde{\Delta\theta}_2 = (-0.4, 0.6)$, $\widetilde{\Delta\theta}_3 = (-0.4, 0.1)$. The server sums them:

$$\sum_i \widetilde{\Delta\theta}_i = (1.3 - 0.4 - 0.4,\ -0.7 + 0.6 + 0.1) = (0.5, 0.0), \tag{18}$$

which equals the true sum of unmasked updates $\sum_i \Delta\theta_i = (1.0 + 0.2 - 0.7,\ -0.5 + 0.1 + 0.4) = (0.5, 0.0)$. Thus, the server learns only the aggregate update and not any individual client update.

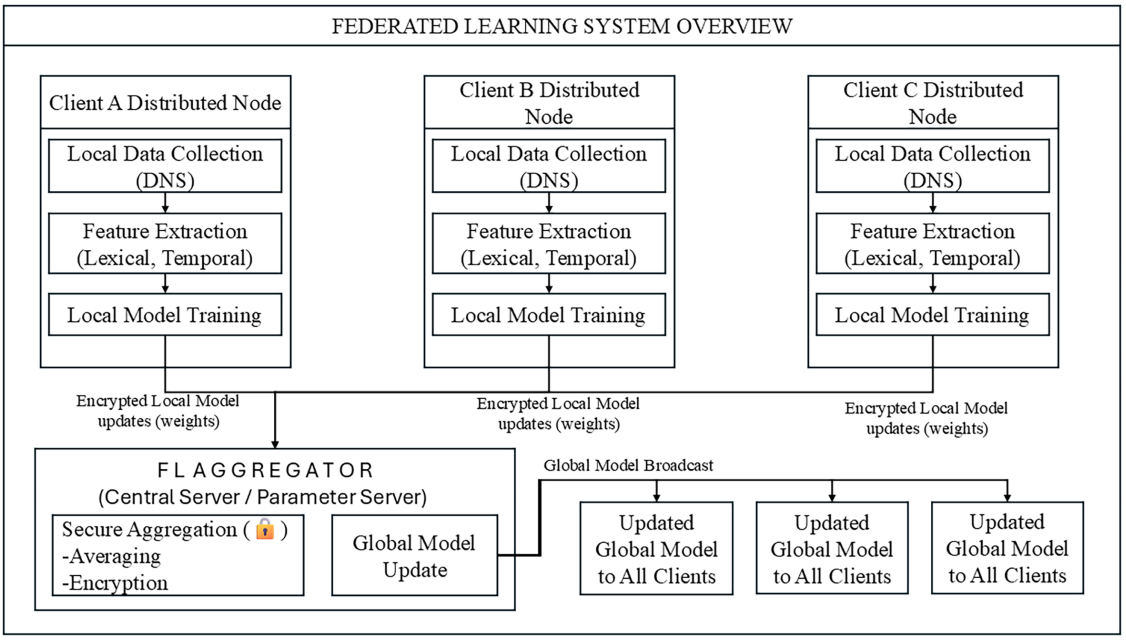

Figure 4: Secure aggregation with DP noise; the server has no access to individual client updates.

5.2 Differential Privacy (DP)

We apply server-side DP by clipping the aggregated update and adding Gaussian noise (Section 3). Across repeated model refresh cycles in long-term DNS monitoring, privacy loss composes over time; therefore, deployments should track a privacy budget per model/version and account for cumulative (ε, δ) across refreshes.

5.3 Poisoning Awareness and Robustness

We screen updates with lightweight anomaly checks (e.g., norm caps and cosine-similarity outliers) and evaluate simple poisoning stress tests (label-flip/sign-flip). Robust aggregation options (median/trimmed-mean) are discussed and included in ablations in Section 6.

6 Experimental Evaluation

This section details the evaluation setup, non-IID regimes, metrics and statistical methodology, hyperparameters, and results. We also report ablations (model choice), privacy/robustness stress tests, and communication accounting.

6.1 Evaluation Setup and Baselines

We compare centralized training, local-only training (per-client models), and three federated optimizers: FedAvg, FedProx, and FedNova. Unless stated, we use the dataset partitions and privacy-compatible features from Section 3. Centralized training pools all *training* partitions only (no test leakage). Local-only trains a separate model for each client and averages metrics across clients on their *own* test slice. Federated runs use the same validation policy and early stopping across methods.

6.2 Non-IID Regimes and Client Participation

We evaluate three heterogeneity regimes using a Dirichlet label skew with concentration α and additional feature skew by TLD/DGA family:

- *Mild*: $\alpha = 10.0$, balanced client sizes; participation $q = 1.0$.
- *Medium*: $\alpha = 1.0$, log-normal client sizes (shape $\sigma = 1.0$); $q = 0.7$.
- *Severe*: $\alpha = 0.3$, strong TLD/family stratification; $q = 0.5$ with client churn rate $c = 20\%$.

For all federated runs we train for $R = 10$ rounds or until validation AUC plateaus over a window of 3 rounds (patience).

6.3 Metrics and Statistical Methodology

We report threshold-free utility (ROC-AUC, PR-AUC) and thresholded metrics (Accuracy, Precision, Recall, F1). The operating threshold is set on the validation set by maximizing F1 or achieving a target recall $r_\star = 0.98$; we state the policy in captions.

6.3.1 Confidence Intervals (CIs)

For each metric m, 95% CIs are computed via non-parametric bootstrap with $B = 1000$ resamples at the *domain* level:

$$\hat{m}^{(b)} = m\big(\mathcal{S}^{(b)}\big), \quad \mathrm{CI}_{95\%} = \big[\hat{m}_{2.5},\ \hat{m}_{97.5}\big], \tag{19}$$

where $\mathcal{S}^{(b)}$ is a bootstrap sample of the test set and $\hat{m}_p$ denotes the pth percentile over $\{\hat{m}^{(b)}\}_{b=1}^{B}$.

6.3.2 Significance Testing

For ROC-AUC we use DeLong's paired test against the centralized baseline; for Accuracy we use McNemar's test on paired predictions:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}, \tag{20}$$

where n_{01} (resp. n_{10}) counts instances misclassified by model A but not B (resp. B but not A). We report p-values and mark $p < 0.05$ as significant.

6.4 Hyperparameters and Implementation

Local optimization uses Adam with learning rate $\eta = 10^{-3}$, batch size $B = 680$, epochs per round $E = 1$, weight decay $\lambda = 10^{-4}$, and dropout $p = 0.3$. FedProx uses proximal coefficient $\mu = 10^{-3}$. FedNova normalizes by effective local step counts a_i. Participation rate is $q = 1.0$ for the default results unless otherwise indicated. Unless otherwise noted, clients train on commodity CPUs; the server runs on a single GPU-equipped node for aggregation and evaluation.

6.5 Learning Curves and Convergence Behavior

Fig. 5 shows representative client-side training loss trajectories; Fig. 6 plots global accuracy vs. federated rounds under FedAvg. Convergence speed is reported as rounds to reach target ROC-AUC 0.9998.

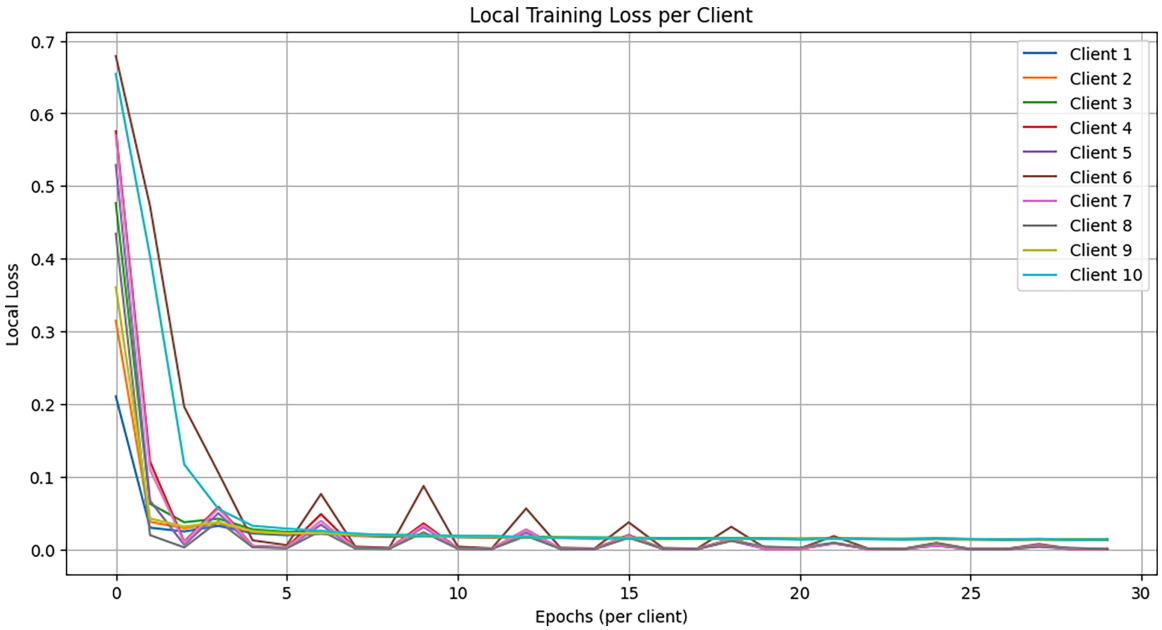

Figure 5: Local training loss per client over epochs (mild non-IID). Shaded regions indicate the interquartile range across clients.

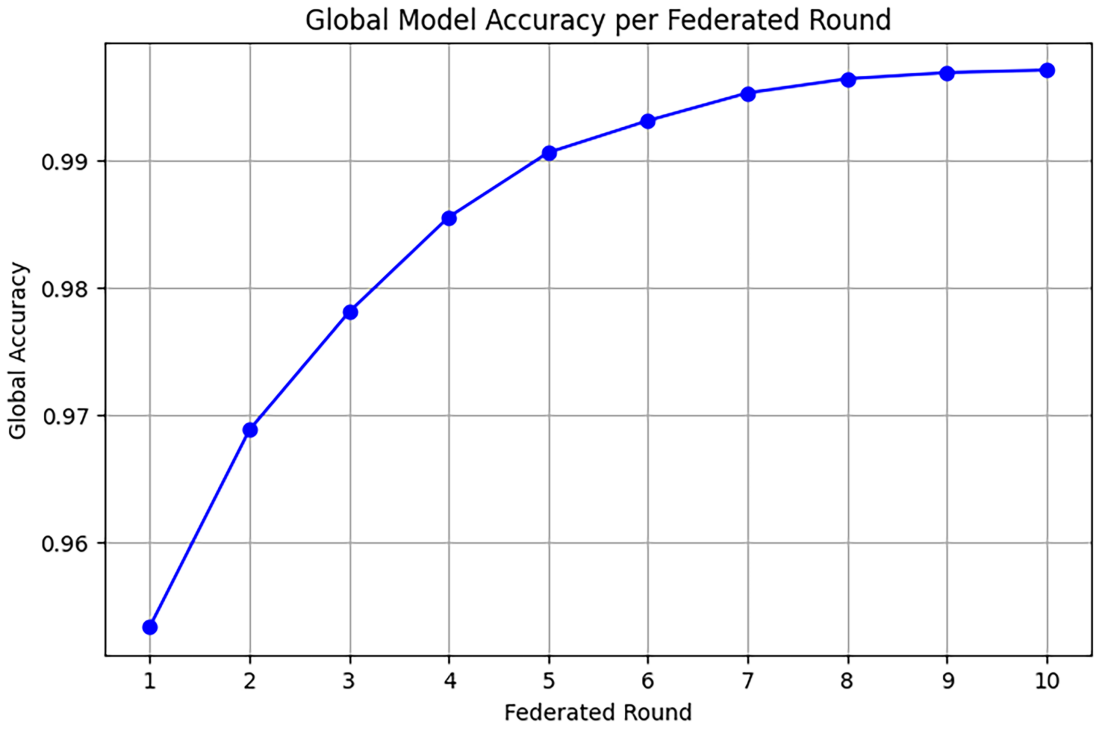

Figure 6: Global accuracy vs. federated rounds (FedAvg) with participation $q = 1.0$. Horizontal dashed line marks the centralized upper bound.

6.6 Main Results

Table 2 summarizes global ROC-AUC and DeLong p-values for clients with Accuracy ≥ 0.995 under FedAvg.

Table 2: Global ROC-AUC and DeLong test p-value for FedAvg compared with the centralized baseline (evaluation restricted to clients achieving Accuracy ≥ 0.995).

Method	Global ROC-AUC	DeLong p-Value (vs. Centralized)
FedAvg	0.99990	$<10^{-12}$*

Note: *The DeLong p-value underflowed to `0.0` in printing; we report a conservative upper bound of 10^{-12}.

6.7 Ablations: Architecture and Loss

Table 3 reports parameter count, client-side latency on a reference CPU, and accuracy metrics for the DNN, a 1D-CNN over character embeddings, and a GRU over tokenized subdomains. The compact DNN typically offers the best accuracy–latency trade–off, while CNN/GRU provide marginal gains at higher cost.

Table 3: Ablation: model choice (DNN vs. 1D-CNN vs. GRU) and client-side latency. Latency is median single-sample CPU time over 100 runs.

Arch.	Params (K)	Latency (ms)	AUC	F1
DNN (ours)	4.2	0.08	0.99984	0.99575
1D-CNN	6.4	0.23	0.99978	0.99494
GRU	12.9	0.77	0.99968	0.99449

6.8 Privacy Accounting and Utility Trade-Offs

We evaluate server-side DP with clipping $C = 1.0$ and Gaussian noise $\sigma = 0.5$, client sampling $q = 1.0$, over $R = 10$ rounds. Accounting yields $(\varepsilon, \delta) = (3.5, 10^{-5})$ for composition across rounds; we report the AUC/F1 delta relative to the non-DP setting and the rounds-to-target increase.

6.9 Robustness to Poisoning

We simulate (i) label-flip on a fraction $\rho = 0.3$ of a selected client's positives and (ii) sign-flip of gradient updates on an adversarial client subset of size 20%. We evaluate three defenses: (a) norm clipping of client updates at $c_u = 2.0$, (b) cosine-similarity outlier filtering with threshold $\tau = 0.7$, and (c) robust aggregation (median/trimmed-mean) in an ablation. We report AUC drop under attack and residual drop with defenses: $\Delta\text{AUC}_{\text{attack}}$ and $\Delta\text{AUC}_{\text{defended}}$.

6.10 Communication and Computation Accounting

Let P be the number of model parameters and b the bits per parameter in the payload (32 for FP32; 8 for int8 quantization if enabled). Per round,

$$\text{Uplink (MB)} \approx \frac{|\mathcal{S}_r| \cdot P \cdot b}{8 \cdot 10^6}, \qquad \text{Downlink (MB)} \approx \frac{P \cdot b}{8 \cdot 10^6}. \tag{21}$$

We also report total traffic to the target AUC and the number of rounds saved by optimizer choice. Table 4 summarizes communication for each federated variant.

Table 4: Communication cost to target under non-IID medium (participation $q = 1.0$). Update size assumes FP32 payloads. Traffic is cumulative across all participating clients. Target is Acc ≥ 0.995.

Variant	Update Size (MB/client/round)	Rounds to Target	Uplink (MB)	Downlink (MB)
FedAvg	0.0166	7	1.165	1.165
FedProx	0.0166	3	0.499	0.499
FedNova	0.0166	−(10 rnds)	1.664	1.664

6.11 Reporting Protocol and Reproducibility

We fix random seeds $\{1, 2, 3\}$ for data partitioning and initialization and repeat each experiment three times; tables report the mean and 95% CIs across repeats. All preprocessing (standardization, threshold selection, temperature scaling) uses training/validation only. Code and partition recipes will be released upon publication, subject to data-governance constraints.

6.12 Summary of Findings

Across non-IID regimes, federated variants approach the centralized upper bound while significantly outperforming local-only training. Under medium/severe skew, FedProx or FedNova reduce rounds-to-target with similar or slightly better AUC/F1. DP at $(\varepsilon, \delta) = (3.5, 10^{-5})$ incurs modest utility loss, and lightweight poisoning defenses recover a substantial fraction of the attack-induced degradation.

7 Discussion and Limitations

This section discusses deployability and operational trade-offs observed in our study, then enumerates explicit limitations and directions for future work. While our federated approach narrows the gap to a centralized upper bound and preserves data locality, several practical hurdles remain before large-scale production adoption.

7.1 Deployment Considerations

Cross-silo deployments involve long-lived clients (enterprises, ISPs, campuses) with heterogeneous resources, policies, and participation rates. In a realistic DNS workflow, each organization runs a local feature-extraction and training service adjacent to its resolver or DNS analytics pipeline. Resolver events are sanitized on-premises (e.g., reducing QNAMEs to registered domains, aggregating to window-level counters, and removing direct identifiers), after which privacy-compatible lexical and behavioral features are computed locally. At an operational cadence aligned with change-control windows (e.g., nightly or weekly), clients download the current global model, train locally for a small number of epochs, and upload *masked* updates for aggregation. Operators can monitor per-round utility and drift indicators and, when necessary, roll back to a prior global checkpoint using standard MLOps practices.

Although cross-silo participants are more stable than cross-device FL, dropouts and stragglers still occur due to maintenance windows, connectivity limits, or policy constraints. We therefore view partial participation ($q < 1.0$) and timeout-based rounds as practical defaults, with late clients deferred to subsequent rounds. In production, federation parameters should be adapted to heterogeneous client capabilities—for example, reducing local epochs E or batch size on constrained clients to mitigate straggling, scheduling uploads during off-peak windows to respect bandwidth budgets, and selecting optimizers that are less sensitive to variable local steps when participation fluctuates.

7.2 Privacy and Governance

Our design keeps raw DNS data on-premises and uses secure aggregation (SecAgg) together with optional server-side differential privacy (DP). SecAgg prevents the server from inspecting individual client updates under an honest-but-curious threat model, while DP bounds information leakage from the aggregated update across rounds under a specified (ε, δ) budget. At the same time, these mechanisms do not prevent an adversarial client from learning from the released global model; therefore, governance controls such as authenticated access, model distribution policies, and audit logging remain essential in operational environments.

We also emphasize the practical privacy-utility trade-off: stronger clipping and noise improve privacy guarantees but may reduce detection utility, especially under severe class imbalance and non-IID skews. For deployment, we recommend selecting privacy budgets via policy requirements first and then validating operating points empirically, reporting both utility metrics and the corresponding accounting parameters (e.g., clipping threshold C, noise scale σ, participation rate q, and number of rounds R).

7.3 Robustness and Security Posture

We considered simple label-flip and sign-flip attacks and evaluated lightweight defenses appropriate for cross-silo operations. The primary attack surface arises from malicious or compromised participants that can manipulate labels, craft poisoned updates, or attempt backdoor behaviors; the risk is amplified in DNS MDD because positives are sparse and client data are non-IID, so a small adversarial fraction may disproportionately affect rare families. As mitigations, we found that screening mechanisms (e.g., norm caps and cosine-similarity outlier filtering) and robust aggregation variants (e.g., median or trimmed-mean) can reduce attack impact with minimal coordination. These defenses primarily add server-side computation rather than bandwidth overhead and can be combined with SecAgg via pre-aggregation screening on client-side statistics or post-aggregation checks on the summed update. Stronger Byzantine-resilient protocols, verifiable computation, or attested execution are promising directions when adversarial participation is higher, but they introduce additional complexity and operational cost.

7.4 Generalization and Drift

Cross-time generalization was evaluated via temporal splits, yet real deployments face seasonal effects, vendor migrations, and emergent DGA families. In practice, shifts in TLD popularity or resolver behavior can reduce precision, so periodic re-federation and drift monitoring are important. We recommend combining routine re-training with distribution-shift alarms based on summary statistics (e.g., population stability index (PSI) and Jensen-Shannon (JS) divergence on key features) to detect when the feature or label prior has moved outside the training regime.

Open-set dynamics remain challenging because newly emerging DGAs and domain-registration patterns are not fully captured by historical engineered features. Incorporating representation learning (e.g., subword/domain embeddings) may improve out-of-distribution tolerance, but it increases model size and may require careful client-side compute and privacy considerations.

7.5 Communication and Compute Trade-Offs

Federation replaces raw-log transfer with parameter exchange. For a model with P parameters and payload precision b bits, per-round downlink is $\frac{Pb}{8\cdot 10^6}$ MB and per-round uplink is $\frac{|\mathcal{S}_r|Pb}{8\cdot 10^6}$ MB. Int8 payloads or sparsification reduce bandwidth but can affect convergence; we leave systematic compression–utility studies to future work. Client compute scales with $E\,|\mathcal{D}_i|\,P$, favoring smaller models for wide deployments.

Operational realism. The communication analysis above uses FP32 as a baseline and full participation in the default setting. In practice, deployments often use partial participation ($q < 1.0$) and may employ quantization (e.g., int8), sparsification, or delta compression to reduce bandwidth. These mechanisms reduce per-round payloads but can interact with optimizer stability and should be evaluated as first-class components in future pilots.

7.6 Limitations

The present study has several limitations that temper external validity. First, portions of the dataset rely on synthetic queries and public threat feeds; although we used temporal holds and cross-time tests, passive-capture coverage is limited and may not reflect resolver-specific caching and TTL behavior. Second, our experiments emulate a modest number of clients with controlled non-IID, so results may change with dozens to hundreds of heterogeneous participants, intermittent connectivity, or highly variable participation schedules. Third, robustness evaluation focused on simple poisoning patterns and lightweight defenses; we did not study adaptive adversaries, stealthy backdoors, collusion among clients, or privacy attacks that combine side information with repeated model snapshots. Finally, we emphasized FedAvg, FedProx, and FedNova; other approaches such as control-variate methods (e.g., SCAFFOLD) and personalization strategies could further improve non-IID performance but introduce additional system complexity. While our feature set was designed to be privacy-compatible, certain aggregations (e.g., IP geodiversity) may still be sensitive in strict environments and could require further coarsening or alternative encodings.

7.7 Future Work

Several extensions can strengthen practicality and scientific value. A key next step is to validate the approach in larger cross-silo pilots with more organizations, longer time horizons, and real resolver logs governed by appropriate data-use agreements. Methodologically, personalization and adaptive weighting (e.g., client-specific heads or meta-learning) may address persistent non-IID while preserving privacy guarantees. From a systems perspective, compression-aware training that combines quantization or sparsification with optimizer choices can reduce bandwidth and rounds-to-target, but should be evaluated jointly with stability and privacy accounting. On the security side, stronger robustness evaluations–including Byzantine-resilient aggregation, integrity mechanisms via secure enclaves/attestation, and defenses tailored to backdoor threats–remain important for adversarial environments. Finally, tighter privacy accounting (e.g., RDP/moment accountants) and representation learning for open-set DGAs are promising, provided client-side compute and governance constraints are satisfied.

In summary, federated DNS malicious-domain detection is feasible with a compact model and a lightweight privacy stack, offering utility close to centralized training while keeping data local. Real-world deployment will benefit from expanded pilots, stronger robustness guarantees, and compression-aware protocols that align with organizational privacy and operational constraints.

Eployment will benefit from expanded pilots, stronger robustness guarantees, and compression-aware protocols that align with organizational privacy and operational constraints.

8 Conclusion

This work investigated federated learning (FL) for malicious domain detection (MDD) from privacy-compatible DNS features across multiple organizations. We developed a compact pipeline in which clients train a lightweight DNN locally and share only masked model updates for secure aggregation, with optional server-side differential privacy (DP) applied to the aggregated update. Across mild to severe non-IID client

regimes, federated models approached the centralized upper bound and consistently outperformed local-only training, while preserving data locality and supporting realistic cross-silo participation patterns.

We further observed that optimizer choice provides a practical lever under stronger heterogeneity: FedProx and FedNova generally reduced rounds-to-target under medium/severe skew with comparable (and in some settings slightly improved) final utility relative to FedAvg. Adding DP at moderate noise levels incurred a modest, measurable cost in ROC-AUC/PR-AUC and F1 that can be controlled through standard accounting parameters (e.g., clipping threshold, noise scale, participation rate, and number of rounds). Finally, lightweight poisoning defenses such as norm caps and cosine-based outlier filtering improved resilience against simple label-flip and sign-flip attacks without increasing communication overhead.

Overall, the results indicate that privacy-preserving FL is a viable alternative to centralized training for DNS-based MDD, delivering near-centralized accuracy while keeping sensitive resolver data on-premises. At the same time, external validity will benefit from larger cross-silo pilots with real resolver logs and longer time horizons, and robustness should be strengthened against adaptive/backdoor attacks and higher adversarial participation. Personalization, control-variate methods, and compression-aware optimization are natural extensions to further balance utility, privacy, and cost at scale.

Acknowledgement: The authors gratefully acknowledge the guidance and support of their supervisor during the completion of this study.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contributions to the paper as follows: study conception and design: Samar Abbas Mangi, Samina Rajper; data collection: Samar Abbas Mangi, Noor Ahmed Shaikh; analysis and interpretation of results: Samar Abbas Mangi, Shehzad Ashraf Chaudhry; draft manuscript preparation: Samar Abbas Mangi, Samina Rajper; manuscript review and editing: Samina Rajper, Noor Ahmed Shaikh, Shehzad Ashraf Chaudhry. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this study is publicly available on Kaggle at: https://www.kaggle.com/datasets/nizamuddinmaitlo/malicious-domain-detection-dataset.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sahoo D, Liu C, Hoi SCH. Malicious URL detection using machine learning: a survey. arXiv:1701.07179. 2017.
2. Zhauniarovich Y, Khalil I, Yu T, Dacier M. A survey on malicious domains detection through DNS data analysis. arXiv:1805.08426. 2018.
3. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. arXiv:1602.05629. 2017.
4. Sheller MJ, Reina GA, Edwards B, Martin J, Bakas S. Multi-institutional deep learning modeling without sharing patient data: a federated learning approach for computational pathology. Sci Rep. 2020;10(1):1–12.
5. Hernandez-Ramos JL, Karopoulos G, Chatzoglou E, Kouliaridis V, Marmol E, Gonzalez-Vidal A, et al. Intrusion detection based on federated learning: a systematic review. arXiv:2308.09522. 2023.
6. Belenguer L, Navaridas J, Pascual JA. A review of federated learning in intrusion detection systems for IoT. arXiv:2204.12443. 2022.
7. Hamad NA, Abu Bakar KA, Qamar F, Jubair AM, Mohamed RR, Mohamed MA. Systematic analysis of federated learning approaches for intrusion detection in the Internet of Things environment. IEEE Access. 2025;13:95410–44. doi:10.1109/access.2025.3574672.

8. Agrawal S, Sarkar S, Aouedi O, Yenduri G, Piamrat K, Bhattacharya S, et al. Federated learning for intrusion detection system: concepts, challenges and future directions. arXiv:2106.09527. 2021.
9. Nguyen VT, Beuran R. FedMSE: federated learning for IoT network intrusion detection. arXiv:2410.14121. 2024.
10. Nivaashini M, Suganya E, Sountharrajan S, Prabu M, Bavirisetti DP. FEDDBN-IDS: federated deep belief network-based wireless network intrusion detection system. EURASIP J Inf Secur. 2024;2024(1):8. doi:10.1186/s13635-024-00156-5.
11. Cajaraville-Aboy D, Moure-Garrido M, Beis-Penedo C, Garcia-Rubio C, Díaz-Redondo RP, Campo C, et al. CO-DEFEND: continuous decentralized federated learning for secure DoH-based threat detection. arXiv:2504.01882. 2025.
12. Chaurasia N, Ram M, Verma P, Mehta N, Bharot N. A federated learning approach to network intrusion detection using residual networks in industrial IoT networks. J Supercomput. 2024;80(13):18325–46. doi:10.1007/s11227-024-06153-2.
13. Almaghthawi A, Ghaleb EAA, Akbar NA, Asiri L, Alrehaili M, Altalidi A. Federated-learning intrusion detection system based blockchain technology. Int J Onl Eng. 2024;20(11):16–30. doi:10.3991/ijoe.v20i11.49949.
14. Doriguzzi-Corin R, Siracusa D. FLAD: adaptive federated learning for DDoS attack detection. Comput Secur. 2024;137(4):103597. doi:10.1016/j.cose.2023.103597.
15. Zhou Q, Mao X, Chen Y. A DDoS attack detection method combining federated learning and hybrid deep learning in software-defined networking. Comput J. 2025;68(10):1463–75. doi:10.1093/comjnl/bxaf049.
16. Elkhawas AI, Chen TM, Gashi I. Privacy-preserving federated learning for phishing detection. IEEE Technol Soc Mag. 2025;44(2):77–84. doi:10.1109/mts.2025.3558971.
17. Qin Z, Yan H, Zhang B, Wang P, Li Y. Real-time identification technology for encrypted DNS traffic with privacy protection. Comput Mater Continua. 2025;83(3):5811–29. doi:10.32604/cmc.2025.063308.
18. Teixeira R, Baldoni G, Antunes M, Gomes D, Aguiar RL. Leveraging decentralized communication for privacy-preserving federated learning in 6G Networks. Comput Commun. 2025;233(4):108072. doi:10.1016/j.comcom.2025.108072.
19. Latif N, Ma W, Ahmad HB. Advancements in securing federated learning with IDS: a comprehensive review of neural networks and feature engineering techniques for malicious client detection. Artif Intell Rev. 2025;58(3):91. doi:10.1007/s10462-024-11082-w.
20. Nkoom M, Hounsinou SG, Crosby GV. Securing the Internet of robotic things (IoRT) against DDoS attacks: a federated learning with differential privacy clustering approach. Comput Secur. 2025;155(4):104493. doi:10.1016/j.cose.2025.104493.
21. Drefahl P, Kostage K, Peppers S, Guo W, Mazzola L, Qu C. Design of a hierarchical federated generative learning based smart home system. In: Digital human modeling and applications in health, safety, ergonomics and risk management. Cham, Switzerland: Springer Nature; 2025. p. 160–77. doi:10.1007/978-3-031-93502-2_11.
22. Zhang Q, Liu M, Li P, Yuan J, Zhu H. Multidomain secure communication and intelligent traffic detection model in VANETs. Int J Intell Syst. 2025;2025(1):2539516. doi:10.1155/int/2539516.
23. Wu G, Wang X. A privacy-enhanced framework with deep learning for botnet detection. Cybersecurity. 2025;8(1):9. doi:10.1186/s42400-024-00307-8.
24. Ndibe OS. AI-driven forensic systems for real-time anomaly detection and threat mitigation in cybersecurity infrastructures. Int J Res Publ Rev. 2025;6(5):389–411. doi:10.55248/gengpi.6.0525.1991.
25. Ali B, Chen G. Next-generation AI for advanced threat detection and security enhancement in DNS over HTTPS. J Netw Comput Appl. 2025;244(3):104326. doi:10.1016/j.jnca.2025.104326.