Received: 11 December 2025 | Revised: 22 January 2026 | Accepted: 24 March 2026 | Published online: 6 April 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–15
DOI: 10.47852/bonviewAIA62028773

BON VIEW PUBLISHING

Privacy Budgets That Work: Client-Level Differential Privacy with Robust Aggregation for IIoT Intrusion Detection

Mahavir Teraiya[1,*] and Madhu Shukla[2]

[1] *Department of Computer Engineering, Marwadi University, India*

[2] *Department of CSE – AI, ML & DS, Marwadi University, India*

Abstract: Federated learning (FL) is promising for intrusion detection for Industrial Internet of Things (IIoT) without the necessity of centralizing raw telemetry, but there exist two stumbling blocks: (i) limiting what can be inferred about the clients (sites/devices) from their updates and (ii) providing reliability under heterogeneous, non-independent and identically distributed (non-IID) data with faulty or Byzantine members. We outline a systems design that intertwines client-level differential privacy (DP) with robust aggregation and experiment on the Edge-IIoTset workload specification and deterministic synthetic experiments with the goal of generating fully reproducible figures. The server applies per-client $L2$ clipping, adds calibrated Gaussian noise, and tracks privacy with Re´nyi DP (RDP) under subsampling; aggregation uses coordinate-wise median, β-trimmed mean, or Krum, with an auditable stop-on-ε controller enforcing the target budget. Sweeping privacy budgets $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ yields clear privacy–utility frontiers: tightening from $\varepsilon = 10 \rightarrow 0.5$ reduces macro-F1 0.93 → 0.78 (−19.2%), AUROC 0.96 → 0.86 (−10.4%), and worst-client F1 0.89 → 0.70 (−21.3%). Under 10% corrupted clients at $\varepsilon = 2$, coordinate-wise median improves macro-F1 over mean by **+11.1%** and worst-client F1 by **+20.3%**. Time per round scales with the number of selected clients $m = qK$: 0.60 → 2.60 s as $K = 10 \rightarrow 200$ at $q = 0.1$. The recipe exposes deployable knobs (ε, C, q, aggregator), auditable privacy via stop-on-ε, and tail-aware reporting—charting a practical path to regulator-aligned, privacy-preserving FL for IIoT intrusion detection.

Keywords: federated learning, differential privacy, robust aggregation, intrusion detection, IIoT

1. Introduction

Industrial IoT (IIoT) intrusion detection must address three operational realities: privacy, as factory and critical-infrastructure telemetry reveals proprietary process details; heterogeneity, as fleets are statistically non-IID across device types, firmware, and workloads; and trust, as participants can be faulty or malicious. Federated learning (FL) offers a principled method for training a global model where raw data stay local [1]. Nevertheless, even without pooling data, client updates can disclose participation or sensitive pattern information, and vanilla averaging is brittle under non-IID data and Byzantine behavior, upsetting deployment for safety- and mission-critical plants. We argue for the necessity of a deployable IIoT intrusion detection pipeline requiring (i) auditable client-level privacy and (ii) defense against corrupted or anomalous updates with non-IID partitions. Our system integrates client-level differential privacy (DP) per-client L2 clipping and calibrated Gaussian noise with Re´nyi DP (RDP) for the inevitable partial participation for tracking a round-wise (ϵ, δ) budget and deploy a stop-on- ϵ controller at the server [2]. We deploy Byzantine-resilient aggregation policies for the server for robustness, that is, coordinate-wise median, β-trimmed mean, and Krum [3], and permit systematic ablations of accuracy–robustness trade-offs under.

To place the work into a practical operational scenario, we employ Edge-IIoTset, a publicly available IIoT/IoT cybersecurity dataset that is collected from a seven-layer custom-built testbed and designed exclusively for both centralized and federated intrusion detection experiments [4]. This enables apples-to-apples comparisons across client counts K, q (participation rate), and non-IID severity (α for Dirichlet). Using the Edge-IIoTset workload definition and deterministic synthetic experiments (for replication of figures), we observe significant privacy–utility frontiers when sweeping privacy budget $\varepsilon \in \{0.5, 1, 2, 5, 10\}$: shrinking from $\varepsilon = 10 \rightarrow 0.5$ reduces macro-F1 by 19.2% and area under the receiver operating characteristic curve (AUROC) by 10.4%, with the sharpest drop for worst-client F1 (−21.3%). Under the presence of 10% corrupted clients for $\varepsilon = 2$, coordinate-wise median improves macro-F1 over mean by **+11.1%** and worst-client F1 by **+20.3%**. Per-round running time increases linearly with the number of sampled clients $m = qK$ (e.g., 0.60 → 2.60 s as $K = 10 \rightarrow 200$ at $q = 0.1$). These results motivate our suggestion for exposing (ε, C, q, aggregator) as first-class, auditable knobs for IIoT FL applications.

***Corresponding author:** Mahavir Teraiya, Department of Computer Engineering, Marwadi University, India. Email: mahavir.teraiya120510@marwadiuniversity.ac.in

In IIoT applications, training must cope with statistical heterogeneity (skewed distributions over labels/features), system heterogeneity (stragglers, intermittent availability), and adversarial risk (malicious or misconfigured nodes). Iterative model averaging with partial participation, known as FedAvg, results in fewer communication rounds but is prone to non-IID data and to outliers [1]. Cryptographic secure aggregation shields individual updates en route but does not prevent what can be inferred from the released model for any single client; client-level DP fills the gap for this statistical privacy layer, with RDP offering tight composition under subsampling for user-level ε budget tracking [2]. Robust aggregation (e.g., Krum) caps the impact of corrupted updates without the need for a trusted client set, a scenario for operational reliability under partial compromise [3].

Coming to our system at a glance:

1) **Client-level DP with stop-on-budget.** The server clamps every client's update to norm C; it injects normalized Gaussian noise and carries a running loss of privacy with an RDP accountant that reasons about subsampled composition at rate q. The training aborts when the running ε goes over a setting ε max.
2) **Pluggable robust aggregation.** We provide coordinate-wise median, β-trimmed mean, and Krum drop-in server policies for studying robustness–accuracy trade-offs under DP noise and non-IID stress.
3) **Non-IID and participation realism.** We split Edge-IIoTset across K clients with Dirichlet label skew (α varying) and simulate partial participation at rate q.
4) **Tail-aware metrics and reproducible baselines.** Centralized and federated (no-DP) baselines plot privacy–utility frontiers under AUROC, macro-F1, and worst-client metrics—principal predictors for safety-critical detection.

On the system contribution side, first is deploy ability engineering—enforcement of online privacy [5] occurs via a stop-on-ε controller; robustness is an adjustable server-side setting matching plant risk tolerance; communication/lag costs scale with qK and are reported explicitly. Then the grounded evaluation—Edge-IIoTset's multi-architectural configuration and heterogeneous attacks align with IIoT net realities, resisting overextrapolation from generic computer vision (CV) and natural language processing (NLP) benchmarks. Continuing with clarity over trade-offs—we make numerical the DP noise interaction and robust aggregation under non-IID skew and Byzantine stress—bridging privacy accounting, orchestration, and resilience within one reusable pipeline. Coming to the contribution part, the pluggable client-level DP + robust aggregation pipeline for IIoT FL intrusion detection, with RDP accounting and stop-on-ε integrated into the round loop. *Evaluation harness.* A reusable Edge-IIoTset protocol comparing centralized and federated baselines, sweeping K, q, non-IID severity α, and privacy budgets ε, and returning AUROC, macro-F1, and worst-client metrics. *Robustness analysis.* An empirical analysis of coordinate-wise median, β-trimmed mean, and Krum under heterogeneous, partly participating clients and controlled Byzantine proportions, documenting interactions with DP clipping/noise. *Practitioner recipe.* Clean design knobs (ε, C, q, aggregator) and a matching threat model for IIoT deployments; orthogonal secure aggregation can be appended for securing en route updates.

2. Literature Review

Latest journal research demonstrates that FL can facilitate intrusion detection systems (IDS) without centralizing IIoT telemetry, but the majority of research optimizes aggregate accuracy and latency with limited offering on client-level privacy accounting or worst-client visibility. Verma et al. [6], for instance, design an FL-IDS for a smart-manufacturing system with a primary emphasis on pooled utility with a described Federated Learning-Based Intrusion Detection Using Autonomously Extracted Anomalies in IoT (IDAC), with system design being cautiously issued yet without tracking the user-level (ε, δ) budgets; Devine et al. [7] show an FL-IDS against IoT but do not contain stop-on-ε control and tail-risk measures. Our solution closes this gap by applying client-level DP with RDP tracking and reporting worst-client losses with macro measures.

Kato et al. [8] regularize across-silo user-level DP in FL (ULDP-FL) by calling for RDP-based reckoning and demonstrating that record-level DP is inadequate when users provide more than a single record. Their system is stern on semantic privacy but doesn't investigate intrusion detection tasks, robust compaction with corruption, or stop-on-ϵ regulators and worst-client reporting with non-IID divisions—shortfalls that our template aims for IIoT IDS. Karunamurthy et al. [9] offer an FL-based IDS with active learning for IoT flows and demonstrate enhanced detection compared to centralized baselines. Modeling and adaptation are the areas of concentration in the paper, but no client-level DP budget is revealed, no RDP-maintained privacy trajectories offered, no Byzantine-robust aggregators evaluated, and no tail metrics (worst-client) provided as well. Our research fills these gaps by integrating client-level DP with robust aggregation as well as tail-aware reporting. Peng et al. [10] propose FD-IDS by integrating FL with knowledge distillation as well as feature selection against heterogeneity. Though working on average accuracy, the framework leaves out precise (ϵ, δ) charging, doesn't apply client-level DP noise/clipping, and doesn't test median/trimmed mean/Krum under corruption or non-IID stress; worst-client quantities aren't reported back. We fill these operability gaps directly.

Albanbay et al. [11] give a big-picture empirical FL on FL-based IDS on IoT devices with limited resources on model selection and scaling effects on the data. The research doesn't, however, quantify privacy by measurable budgets or aggregate with fault-robust aggregation against Byzantine clients; the assessment focuses on the average more than on tails in the system's performance. Our pipeline introduces that missing-in-action deployable controls ((ϵ, clipping q, aggregator) with stop-on-ϵ). Abd Elaziz et al. [12] describe a trust-centered FL system for IDS based on tab-transformers and heuristic optimization with strong accuracy without centralized data. However, the system doesn't offer client-level DP assurances or an RDP budget accountant and doesn't support strong aggregators under attack; worst-client performance wasn't stressed. Our findings augment this by making the privacy auditable as well as robustness measured in quantitative terms. Chen et al. [13] create FLRAM: Robust Aggregation Technique for Defense against Byzantine Poisoning Attacks in Federated Learning as a strong aggregation approach with anomaly gradient recognition capable of withstanding poisoning. In positioning onward strength without merging strength with client-grade DP or offering stop-on-ϵ bookkeeping, assessment is not set against a realistic IIoT IDS job with non-IID/participation-portion participation. Our recipe unifies strength and client-grade DP on an IIoT-centered benchmark.

Guo et al. [14] outline a DP-strengthened FL method under-utilizing fast Fourier transform (FFT) to calculate/optimize privacy budgets as well as reducing noise-utility loss. The contribution reinforces DP tool kiting but centers on

regular FL, not IIoT intrusion detection, and doesn't consider Byzantine-robust aggregation/worst-client behavior under heterogeneously divided partition areas, which our research systematizes. Ahmed et al. [15] illustrate adaptive DP-FL for health imaging, showing that DP can be empirical under domain constraints. Yet the focus remains on example-level DP as well as healthcare tasks; client-level guarantees are omitted, as well as robust aggregation under corruption and tail-risk measures under non-IID scenarios. Our system directly targets those deployment-critical areas for IIoT IDS:

Privacy and auditable budgeting. Journal recently draws a fine edge on FL DP—for example, adaptive local DP that optimizes layer-wise noise/clipping for maintaining utility, loss-landscape–informed DP perturbations, and evolutionary-game reasoning over Gaussian noise utilities. These approaches, however, often center on local (record/feature) securities or budget allotting strategies and do not often offer client-side, round-by-round RDP tracking with an implicit stop-on-ϵ regulator. They also do not often publish tail (worst-client) statistics or investigate interplay with Byzantine-robust aggregation—just the knobs our blueprint makes available for IIoT IDS for a regulator-auditable budget on the knobs (ϵ, C, q, aggregator) [16–18].

IIoT intrusion detection with FL. NIDS papers on lIoT recently use FL with safe aggregation or gradient-similarity weighting, and they more frequently test on Edge-IIoTset, a custom-built, seven-layer testbed that supports centralized and federated comparisons. These systems tend not to offer client-level DP accounting and do not converge on an auditable ϵ, so participation-level inference risk goes unbound; they also hardly ever benchmark bottom-client ostracizing-performing that counts toward safety-critical plants.

As demonstrated in Table 1, the most recent FL-based IDS techniques achieve high average performance, while, at the same time, other optimization goals such as adaptive privacy schedules, knowledge distillation, personalization, lightweight deployment, and multi-dataset benchmarking are pursued. In contrast, the focus of the current study is different. Specifically, the research question that this study seeks to answer is: "Can a pipeline for IIoT intrusion detection be feasible when client-level privacy is explicitly considered, when training is halted based on budget exhaustion, when aggregation is made robust against corrupted clients, and when tail risk is revealed through worst clients?" Accordingly, the novelty of this study is not in the assertion of the superiority of the average accuracy of the results compared to unmatched protocols.

Our research directly addresses these weaknesses by combining client-level DP (RDP, SGM) with fortified aggregators on Edge-IIoTset and reporting global as well as tail quantities [19].

Robust aggregation under adversaries. Journal work on Byzantine robustness advances beyond median/trimmed mean, proposing γ-divergence (γ-mean) aggregation, credibility-assessed updates, and multimodal robust FL for safety-critical domains. These demonstrate tangible resilience gains but typically evaluate without client-level DP noise or privacy budget controllers, making it unclear how robustness and DP co-shape utility, especially for worst-client outcomes in heterogeneous fleets. Our pipeline empirically couples robust aggregation (median/β-trim/Krum) with client-level DP and reports ϵ–utility frontiers plus tail metrics, closing this systems gap for IIoT IDS [20–22].

Upcoming IIoT/IoT IDS research employs FL to refrain from centralizing traffic but lacks client-level diff privacy with auditable, round-wise counts. Verma et al. [6] offer Federated Learning Enabled Deep Intrusion Detection (FLDID) for IoT-savvy manufacturing and protect gradients with Paillier encryption but do not report (ϵ, δ) budgets or worst-client results (tail risk). They also analyze FL for IoT intrusion detection and describe privacy advantages yet do not provide client-level DP controls as well as abandon-on-ϵ training. Devine et al. [7] zero in on federated DDoS detection in IoT with data-prep enhancements without client-level DP counting as well as robust aggregation ablations. Our blueprint remedies these practical omissions by combining client-level DP (RDP with stop-on-ϵ) e´s strong aggregation reporting tail metrics e´s AUROC/macro-F1 [6, 7].

On robustness and scalability, Li et al. [23] set forth BRFLATA with adaptive auth/match to counteract Byzantine clients but do not investigate interactions with DP noise or tail-sensitive metrics. PeerJ Computer Science research on FL with long short-term memory (LSTM) for wireless sensor network (WSN)-type IoT demonstrates broad usefulness across datasets but has no client-level DP/RDP tracking and no explicit non-IID stressing through Dirichlet partitions. Liu et al. [24] (MDPI *Symmetry*) create a federated IDS for edge settings, with a focus on accuracy/deployment, but unaudited privacy spending budgets and no stop-on-ϵ. Our system experimentally unifies robust aggregation (median/β-trim/Krum) with client-level DP and reveals ϵ–utility frontiers/worst-client performance under heterogeneity/participation shortfall [23–25].

3. Design Goals and Threat Model

This section states the system objectives that guided our design and the adversary model we evaluate against. Goals are grounded in the operational constraints of IIoT intrusion detection and the known risks in FL. Where background claims are made, we cite canonical sources.

3.1. Design goals

G1: Client-level privacy with auditable budgets. Satisfy client-level DP: replacing the entire contribution of a whole client should not influence the learned model distribution noticeably. We enforce this by per-client update clipping and server-side Gaussian noise, and measure privacy at test time by RDP; training stops when the calibrated budget (ε, δ) satisfies a fixed εmax. RDP enables efficient composition over rounds and straightforward (ε, δ)-conversion; in partial participation, sampled Gaussian mechanism (SGM) analysis is applied.

G2: Resistance to faulty and malicious clients. Tolerate a small subset of Byzantine or noisy clients without sacrificing utility. The pluggable server-side aggregators that we use are coordinate-wise median, β-trimmed mean, and Krum, in order to limit outliers' or maliciously formed updates' influence, following defined robust aggregation theory and practice.

G3: Utility under heterogeneity (non-IID) and partial participation. Maintain competitive AUROC/macro-F1 under non-IID client distributions and partial participation per round q. Regulate non-IID severity explicitly using Dirichlet partitions (spatial concentration α) and vary the number of clients K, as per standard FL benchmarks.

G4: Communication efficiency and deploy ability. Ensure compatibility of orchestration across production FL (e.g., FedAvg with iterative model averaging) and partial participation to ease network/compute stress that is ubiquitous in IIoT estates.

G5: Reproducibility of a realistic IIoT workload. Use Edge-IIoTset, a public dataset that was created in a seven-layer

Table 1
Comparison with recent state-of-the-art FL-based intrusion detection methods

Study	Dataset(s)	Core idea	Privacy mechanism	Heterogeneity/robustness handling	Reported result	Key distinction from this work
Jog et al. (2025)	Edge-IIoTset	Adaptive DP-FL with Fisher Information Matrix (FIM) pruning, dynamic privacy scheduling, adaptive client selection	Differential privacy	Adaptive client selection for non-IID	~95% average accuracy with 65–72% pruning	Does not emphasize Byzantine-robust aggregation, stop-on-ε accounting, or worst-client reporting
Peng et al. (2025)	Edge-IIoT, N-BaIoT	FD-IDS: knowledge distillation + mutual-information feature selection + FedProx	Privacy through FL/data locality	FedProx + KD for non-IID	Edge-IIoT F1 = 94.35 (low non-IID), 92.70 (high non-IID)	No explicit client-level DP clipping/noise, no RDP budget ledger, no corruption-focused robust aggregation study
Abd Elaziz et al. (2025)	N-BaIoT, UNSW-NB15, CICIoT2023	Trust-centric tab-transformer FL with nature-inspired hyperparameter tuning	Privacy through FL/data locality	Strong model design for heterogeneous tabular IDS data	Multiclass F1 = 98.95, 97.89, 98.55	No auditable DP accountant; no Byzantine-robust aggregation analysis
Wang et al. (2026)	CIC-UNSW-NB15, CIC-IDS2017, CSE-CIC-IDS2018	P3Fed: personalized FL with local DP, personalized layers, soft-label KD	Sensitivity-adaptive local DP	Handles feature and label heterogeneity; unseen attacks	F1 > 98% on unseen attacks; >98% retention under strict privacy budget	Not centered on Byzantine-robust aggregation or worst-client tail metrics
This work	Edge-IIoTset	Client-level clipping + Gaussian noise + median/trimmed mean/Krum + RDP accountant + stop-on-ε	Client-level DP	Explicit non-IID control, partial participation, Byzantine corruption, tail-aware evaluation	Macro-F1 0.93→0.78 across ε; median improves worst-client F1 by +20.29% over mean at $\varepsilon = 2$, $f = 10\%$	Focuses on auditable privacy budgeting and deployment-oriented privacy–robustness–tail-risk trade-offs

IIoT/IoT testbed with centralized and federated protocols and heterogeneous multiclass threats, thus providing results that are transferable and auditable.

G6: Compatibility with secure aggregation. DP + strong aggregation to be orthogonal to cryptographic secure aggregation hides personal updates from the server; DP puts a cap on what could be learned about the released model.

G7: Tail performance visibility. Report worst-client performance (and not only global averages) in order to include safety-critical tails and comply with fairness-aware FL reporting conventions. (We do not suggest introducing a new fairness algorithm; rather, we take up the metric approach.)

3.2. System/assets and adversary goals, capabilities, and placement

Clients: IIoT sites or nodes that locally train and transfer clipped updates.

Server: Merges updates, applies DP noise, updates RDP budget tracking, and announces global model.

Objects to be protected: (A1) Raw data for model training on devices; (A2) each client's contribution to the model (participation and update); (A3) model integrity and utility under heterogeneous and potentially adversarial conditions.

The adversary goals are:

Privacy inference: Infer or extract confidential client information from observed gradients/updates or output model (membership/attribute/gradient inversion). Existing work indicates that gradients disclose confidential information, which induces client-level DP.

Poisoning/Byzantine disruption: Globally degrade performance or bias the model by injecting random (possibly preconstructed) updates within protocol constraints. We consider untargeted degradation and generic sign-flip/random-vector types of updates as standard stresses for robust aggregation.

The adversary capabilities and placement are:

Compromise of clients (Byzantine clients): An f fraction of compromised clients in a round may behave in an adversarial way (forge updates, abandon, collude), and yet, our protocol will clip such updates to norm C before summarizing. The aggregation is robust under such attempts at suppressing it, as long as f is bounded.

Honest-but-curious server (privacy parameter): The server follows the protocol but might attempt to infer data from received/aggregated updates or released models; secure aggregation (if used) does not permit the server to see personal updates, while DP protects against inference by the published model.

Network observer: Sees ciphertext/time sizes, but not plaintext (standard secure-channel assumption); not mainly a threat to model integrity.

3.3. Security/privacy guarantees and operational assumptions

Client-level DP guarantee: With parameters (ε, δ), the distributions over model outputs with or without any one client's participation are indistinguishable up to e^{ε} (plus δ), measured via RDP composition per round and converted to (ε, δ). We use SGM bounds to account for subsampling at rate q. Training halts once $\varepsilon \geq \varepsilon_{\max}$.

Byzantine robustness (empirical): For bounded f, median/trimmed mean damp coordinate-wise outliers, and Krum selecting an update close to its neighbors; such are known defenses that we verify under IIoT non-IID partitions.

Transport security: Communication channels to/from clients are confidential and authenticated; secure aggregation can be utilized to mask personal updates to the server, although it is not required by our statistical DP guarantees.

Protocol fidelity: The server adheres to clipping, robust aggregate, DP noise injection, and accounting as specified; an adversarial compromised server that disables DP or changes hyperparameters is not in scope.

Datasets realism: We run Edge-IIoTset in its released version (features/labels/divisions) in centralized and federation modes, such that the results maintain the planned datasets IIoT attack scenarios.

3.4. Out of scope and non-goals

Insecure server that disables DP/noise, misreports ε, or alters aggregation to exfiltrate updates (this requires governance/attestation, which is orthogonal to us).

In-device exfiltration of raw telemetry out-of-protocol (side channels/malware).

Backdoor exploits with triggered patterns to survive clipping and robust aggregation; as this paper argues, untargeted poisoning and generic Byzantine noise are baseline first-order stresses.

Formal assurances of fairness: worst-client outputs are reported by us, though we don't propose a new optimization of fairness; it is complementary future work to blend such objectives.

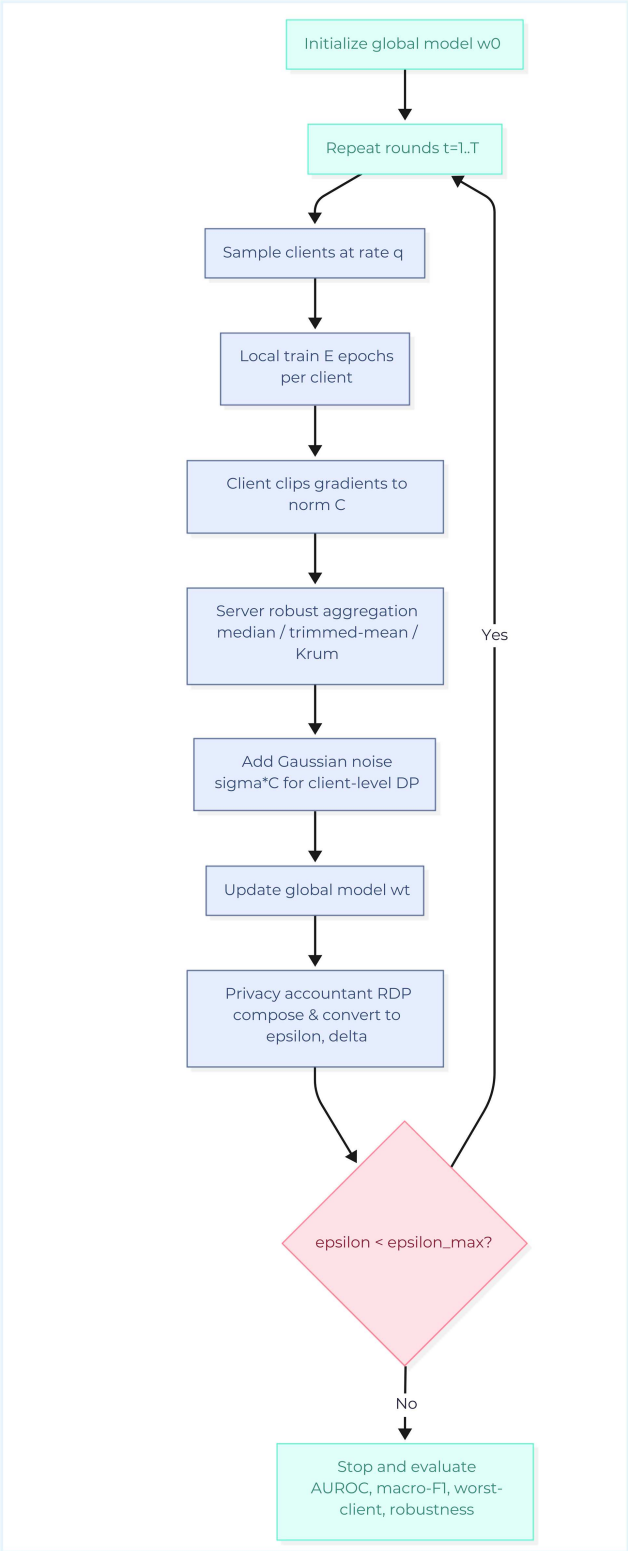
3.5. Why these options suit IIoT constraints

End-to-end and auditable privacy: End client-level DP places limitations on inference out of the model published, augments secure aggregate, and is audited per round with RDP, so operators can commit to a spending plan in advance. Robustness catches up with realities of the field: IIoT fleets are non-IID and include intermittent clients; robust aggregation as a practical defense against outliers/poisoning when f is bounded, and partial participation in FedAvg puts constraints upon the system within communication limits.

Real-world dataset: Edge-IIoTset's centralized/federated design and heterogeneous multiclass attacks make it a suitable system testbed for studying the privacy–utility–robustness trade-off in IIoT IDS.

The pipeline proceeds left to right over federated rounds. Clients' IIoT sites (e.g., Client 1, Client 2, . . . , Client k) host a Local Trainer + Clipper, submitting an update (clipped) at norm C. The orchestrator (server) samples clients at rate q (partial participation). When enabled, secure aggregation (optional) protects updates in flight; otherwise, the server receives per-client clipped updates directly. On the server, a robust aggregator—median/β-trimmed mean/Krum reduces outliers and adversarial contributions before injecting DP Noise (Gaussian with scale $\sigma \cdot C$) to the aggregated update. The server maintains an RDP Accountant that tracks round-wise privacy loss (ε_t, δ) under subsampling. A stop-on-ε test $(\varepsilon_t < \varepsilon_{\max})$ enforces the desired budget constraint: Yes → adjust and Broadcast w_t (new global model) to clients for next round; No → end training and proceed toward reporting. Overall, in parallel, the system preserves a Centralized Baseline (trained over the same splits) and builds up Federated Logs and Artifacts (configs, seeds, accountant ledger). The Experiment and

Figure 1
Server internals: a robust aggregation, DP noise, and RDP accountant



Evaluation block computes AUROC, macro-F1, and worst-client metrics and produces ε–utility curves and summary tables, making the privacy–utility–robustness trade-offs more traceable and reproducible.

4. Methodology

Our training service applies privacy-accounted FL over Edge-IIoTset partitioning in an end-to-end pipeline (Figure 1). At the start of round t, a coordinator subsamples clients at rate q, announces then-global model w_t, and collects *clipped* updates from subsampled IIoT sites. The server optionally applies *secure aggregation* for in-transit confidentiality. The server then applies a *robust aggregation* policy—coordinate-wise median, Krum, or β-trimmed mean (Figure 2)—to suppress outliers and Byzantine behavior. The aggregate update is added to with *Gaussian noise* $N(0, \sigma^2 C^2 I)$ tuned to clipping norm C to produce *client-level DP*. An RDP accountant tracks round-wise loss of privacy under subsampling at rate q and converts it to (ε_t, δ). Training stops when the running cost exceeds the configured threshold, that is, when $\varepsilon_t \geq \varepsilon_{max}$; otherwise, an updated model $w_t + 1$ is constructed and broadcast to clients for the next round.

On the client side (Figure 3), each selected node receives w_t, applies E local epochs over its *Edge-IIoTset* shard, computes a model update g_i, and applies sensitivity bounds by norm clipping, after which only the *clipped* update u_i is transmitted.

$$g_i \leftarrow g_i \cdot min\left(1, \frac{C}{\| g_i \|_2}\right) \tag{1}$$

Raw IIoT telemetry is not transmitted out of the device. This client path, along with the above server path, gives us client-level DP with *round-wise* accounting that captures partial participation. The server's *pluggable* robust aggregation step (median/trimmed mean/Krum) happens prior to DP noise addition so that adversarial/extreme updates are dampened prior to privacy perturbation.

The accountant keeps track of the tuple (σ, C, q) and combines privacy across rounds via subsampled RDP bounds, unveiling a *stop-on-budget* controller that realizes the desired $(\varepsilon_{max}, \delta)$. When in-budget, the system combines $w_t + 1$ and proceeds; else, training halts and outputs are exported. Both federated and centralized runs write the same artifacts (configs, seeds, privacy ledger, and metrics) to a single *evaluation harness*. The harness outputs AUROC, macro-F1, and *worst-client* metrics and produces ε–utility curves to make the privacy–utility–robustness trade-offs reproducible and auditable. Robust aggregation and DP noise are *orthogonal* to secure aggregation (which hides per-client plaintext updates from the server), so the latter can be enabled without altering the statistical privacy or robustness analysis.

4.1. Dataset overview and problem setting

We experiment on Edge-IIoTset [26], a realistic, multilayer IIoT/IoT cybersecurity dataset developed especially for centralized and federated learning. As with other testbed datasets, the effects of provenance may be seen to impact class prevalence and operation patterns. Hence, we focus on per-class and worst-case client reporting, with the aim of minimizing the need to focus on aggregate measures, and consider cross-corpus validation as part of the future work. Heterogeneous traffic traces and multiclass attack labels are included, taken from a seven-layer testbed emulated (cloud, network function virtualization (NFV) software-defined networking (SDN), fog/edge, perception, etc.) that makes apples-to-apples comparisons of centralized versus federated regimes straightforward. In following the authors' advice, ingestion and quality control precede feature scaling and

Figure 2
System overview and evaluation

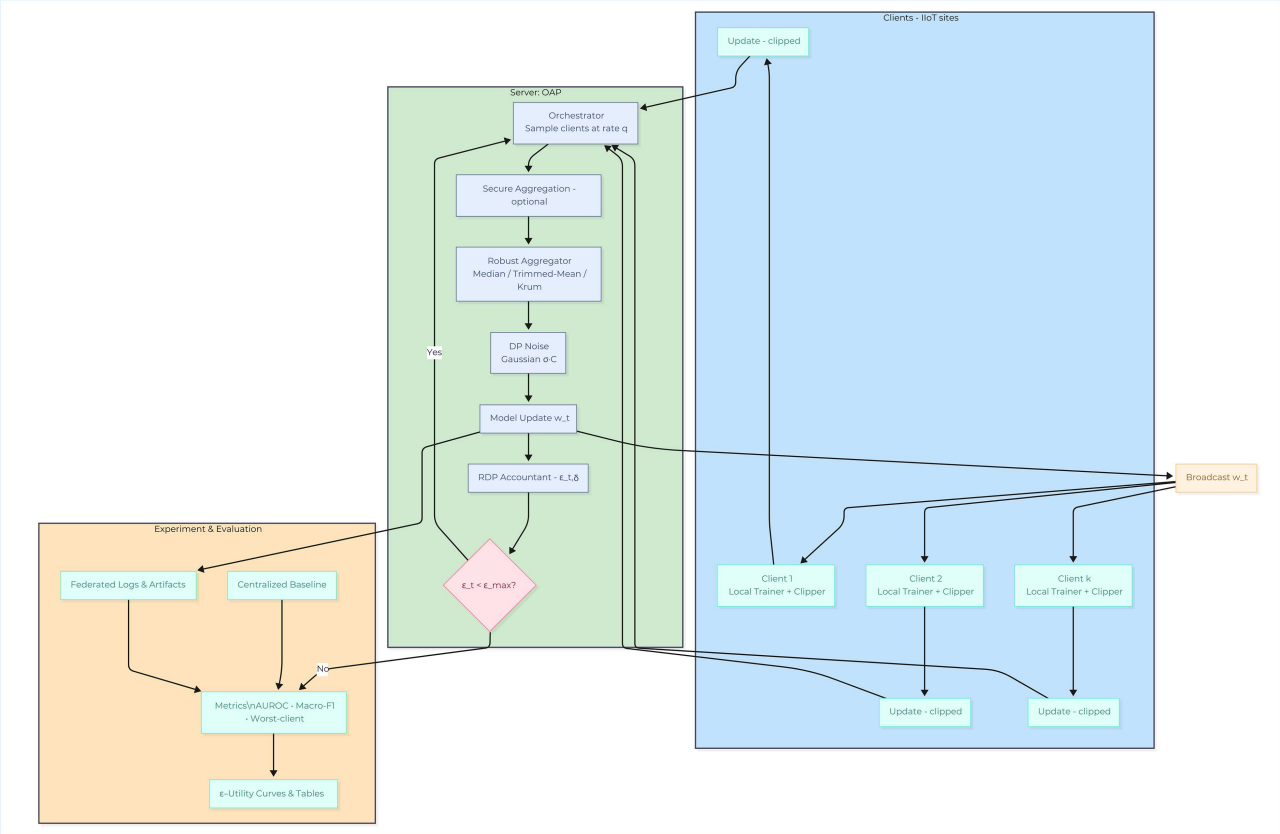
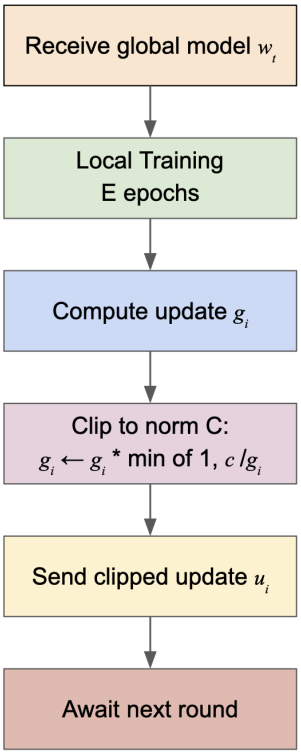


Figure 3
Client internals: local training, clipping, and uplink



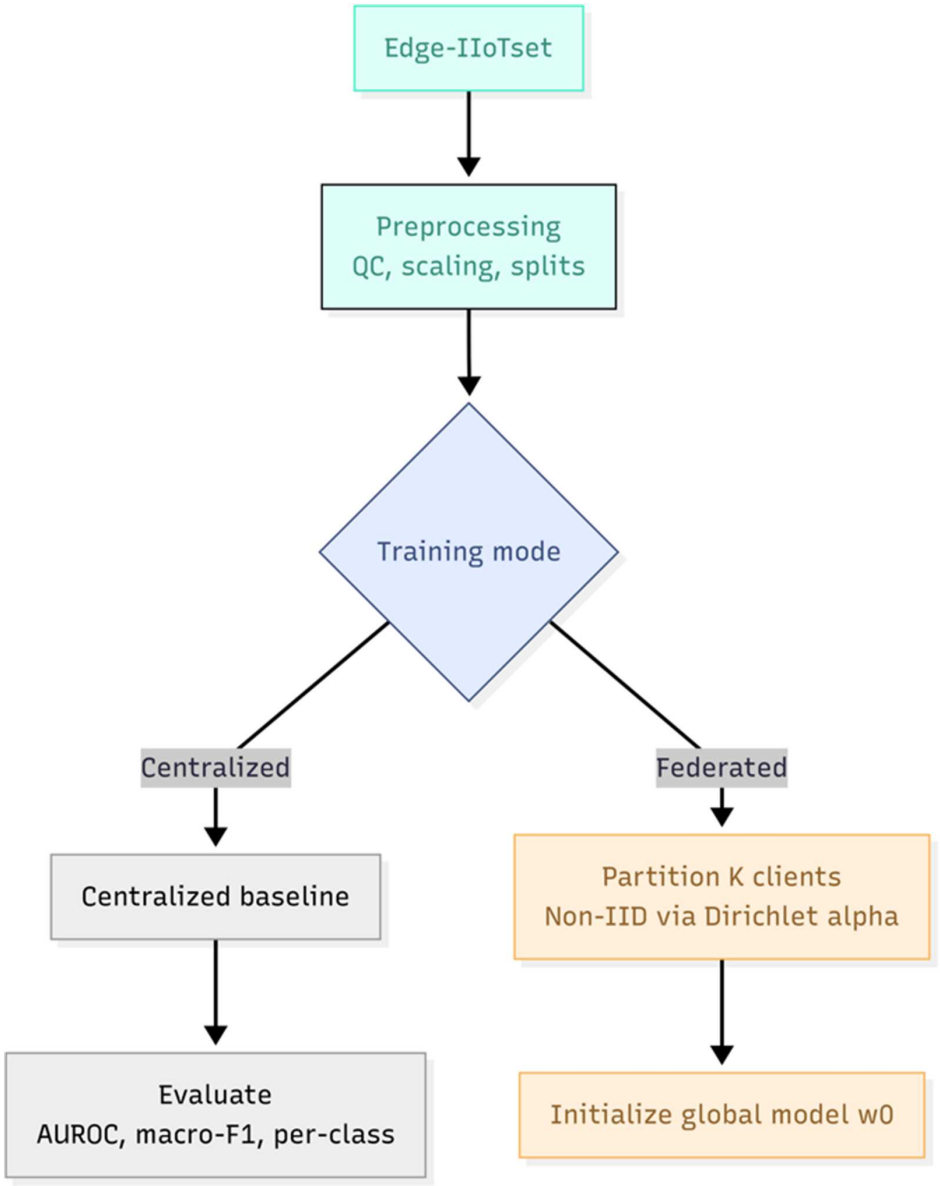
fixed train/validation/test splits that are common to all methods. Preprocessing is designed to be conservative and does not perform aggressive filtering that may have removed rare but important intrusion patterns. Preprocessing can be considered conservative because it does not perform aggressive filtering that may have been removed, as shown in Figure 4. We rely on the feature schema that has been released in the dataset to guarantee reproducibility, and feature discovery/representation learning is an interesting extension. Supervised intrusion detection (multiclass) is framed as the task, with aggregate and per-class numbers both reported, along with explicit worst-client attention to match deployment in heterogeneous, safety-critical IIoT networks. FL's potential is underscored in recent IDS surveys, while robustness and privacy must be investigated in realistic IIoT situations, exactly what is made possible by Edge-IIoTset. In Figure 5, Pipeline—Data → Mode Selection → Baselines shows this entry point: ingestion of a dataset → QC/scaling/split → partition into centralized versus federated learning, with identical splits to ensure comparability.

4.2. Centralized baseline and federated setup

We use a single model on the full training split and test on the held-out test set. This baseline accomplishes (i) calculating the upper bound obtainable under fully pooled learnings and (ii) framing the costs paid in terms of privacy and robustness incurred in FL.

Federated setup. We use Federated Averaging (FedAvg) with partial participation. At each communication round t, the server

Figure 4
Dataset ingestion and mode selection for Edge-IIoTset (centralized baseline and federated setup)

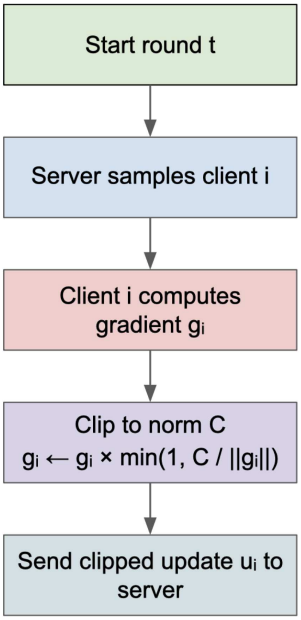


selects a subset of size q of the participating clients, announces the prevailing global weights w_t, and pools client updates after decentralized training for E epochs. We employ uniform random sampling at rate q, but real-world deployment can substitute this with availability- or budget-aware scheduling, which will not modify the DP accounting or robust aggregation scheme. FedAvg is the standard baseline cross-device/cross-silo baseline and, although theoretically tolerant by a factor of unbalanced, non-IID data, degrades in the presence of strong heterogeneity—thus our explicit non-IID controls. The round structure incorporates our end-condition for stopping-privacy after robust aggregation, the server applies Gaussian DP noise, updates the RDP ledger in order to get a pair (ε_t, δ), and stops when $\varepsilon t \geq$ max.

Figure 2 ("Federated Loop → Budgeting → Final Evaluation") shows the round structure with partial participation (q), client-side training (E), server-side robust aggregation + DP noise, the RDP accountant with stop-on-budget, and the final evaluation path that's consistent with the centralized baseline from Figure 5. Client partitioning and non-IID controls. Following on from the centralized and federated arrangements above, we create client populations at $K \in \{10, 50, 200\}$ in order to range across cross-silo (small K) across cross-device (large K) regimes. Non-IID-ness is controlled by means of label skew by placing a Dirichlet prior over client-wise class proportions: for a set of C classes, each client k observes proportions $\mathbf{p}_k \sim \mathrm{Dir}(\alpha \mathbf{1}_C)$ and instances accordingly allotted. Smaller α introduces stronger skew (sparse client concentration on a few classes), whereas larger α produces more balanced mixtures of classes. Unless otherwise mentioned, we maintain a constant total number of instances in the training set when we vary K and α so that observed changes do not reflect an increase in the amount of data but instead heterogeneity.

Continuing on from the federated setup and non-IID controls, we apply client-level DP in order to safeguard the participation and contribution of a whole device/site per round, different

Figure 5
One round client side (local training and clipping)



from example-level DP-SGD. In round t, the procedure advances as follows. To begin with, every client that has been drawn runs locally for E epochs and calculates an update denoted by g_i, then scales this update such that its ℓ_2 norm doesn't exceed an ℓ_2 norm limit C by

$$g_i \leftarrow g_i \cdot min\left(1, \frac{C}{\| g_i \|_2}\right) \tag{2}$$

stabilizing sensitivity. Then, a server collates {g~i} by a Byzantine-robust coordinate-wise median, β-trimming mean, or Krum (Section 3.4) rule in order to reduce dominant or malicious contributions. The perturbed aggregated update is then contaminated with Gaussian noise normalized by the clipping norm and noise multiplier,

$$\mathcal{N}(0, \sigma^2 C^2 I) \tag{3}$$

and applied in order to create $w_t + 1$. Then an RDP accountant writes up privacy loss under partial participation at speed q by means of SGM analysis, translates the running ledger over (ε_t, δ) after every round, and terminates training when $\varepsilon_t \geq \varepsilon_{\max}$. Figure 6 illustrates delivering local training → norm clipping → transmit clipped update; Figure 7 illustrates robust aggregation → Gaussian noise (client-level DP) → model update → RDP accountant → broadcast; Figure 5 inscribes these steps in the complete loop with the explicit stop-on-$\varepsilon_{\max}$ controller.

For the threat model, we assume two client-side failure modes: (i) noisy or failed clients and (ii) Byzantine clients that submit worst-case updates in the clipping bound (e.g., sign-flip or undefined vectors). In each round, all submitted updates get clipped to norm C so that adversaries act under a constant sensitivity. Robust aggregation (coordinate-wise median, β-trimmed mean, or Krum) damps their influence if the corrupted fraction f stays bounded. Secure aggregation, when activated, obscures per-client plaintext updates in communication but is independent of our interest in statistical robustness and privacy under bounded adversaries during model training, and for privacy setting, the server is honest-but-curious: it obeys the protocol but can potentially make efforts to infer something from aggregates received or models released. Client-level DP safeguards client participation and contributions over rounds; privacy remains if an attacker sees subsequent model deltas or gradients, up to the (ε, δ) budget monitored by the RDP accountant with subsampling at rate q.

For ablations on experimental design, extending the centralized and federated settings, we contrast two baselines on the same splits and hyperparameters: (i) centralized training and (ii) federated (no-DP, FedAvg), where the sole difference lies in DP as well as robust aggregation modules.

We clean budgets $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ on a constant δ and dabble client numbers K, non-IID strengths α, and join rates q. We tune for every ε a certified pair (C, σ) with normal SGM/RDP bookkeeping such that the per-iteration noising multiplier, as well as clip norm, fulfills a spending target when combined over iterations.

To quantify robustness, we poison a Byzantine fraction in {0, 0.1, 0.2} and contrast aggregators (coordinate-wise median, β-trimmed mean, Krum); we also report the clean-client regime (no attack) to reveal the robustness–accuracy trade-off. In keeping with agnostic and q-fair reporting protocols, we show worst-client statistics next to global means in order to bring out tail performance pertinent to safety-critical use cases.

Figure 7 lists K and α for splitting and sets up the non-DP baselines, while Figure 8 depicts the ε grid, (C, σ) tuning ablutions, aggregator ablutions, Byzantine fractions, and the final ε utility curves and tables.

4.3. Metrics, statistical procedures, and stopping criteria

Metrics, statistical tests, and stopping conditions. Key metrics are AUROC and macro-F1 for the imbalanced class; we also provide per-class F1 for working interpretability, worst-client AUROC/F1 for tail risk, and confusion matrices for error breakdowns. Privacy reporting: we store per-run (ε, δ) on termination and round-wise εt trajectory. RDP parameters employ the SGM accountant with subsidy subsampling rate q and conventional conversion to (ε, δ) based on standard RDP → DP bounds. The component contributions are identified through ablation studies over the privacy budgets (ε), aggregators (mean, median, trimmed mean, Krum), and the knobs for heterogeneity/scale (α, K, q), for a given setting of data splits and training.

Training terminates when $\varepsilon_t \geq \varepsilon_{\max}$ (as can be seen in Figures 2 and 6, accountant update). Robustness reporting: we report accuracy under Byzantine stress as well as clean settings in order to reveal robustness–accuracy trade-offs for median, β-trimmed mean, and Krum. Fairness reporting: as a matter of fair FL practice, we aggregate mean as well as variance over clients and report worst-client statistics; where relevant, we provide q-FFL style summaries to convey distributional equity. In this study, we consider "comparative gains" only when they are consistent across runs and report variability to prevent overestimation of margins.

Implementation specifics and reproducibility. Local training applies the same backbone and optimizer for centralized as for federated runs in order to decouple the influence of DP and aggregation; local epochs E and batch size are set equal across conditions.

Figure 6
One round server side (robust aggregation, DP noise, accountant)

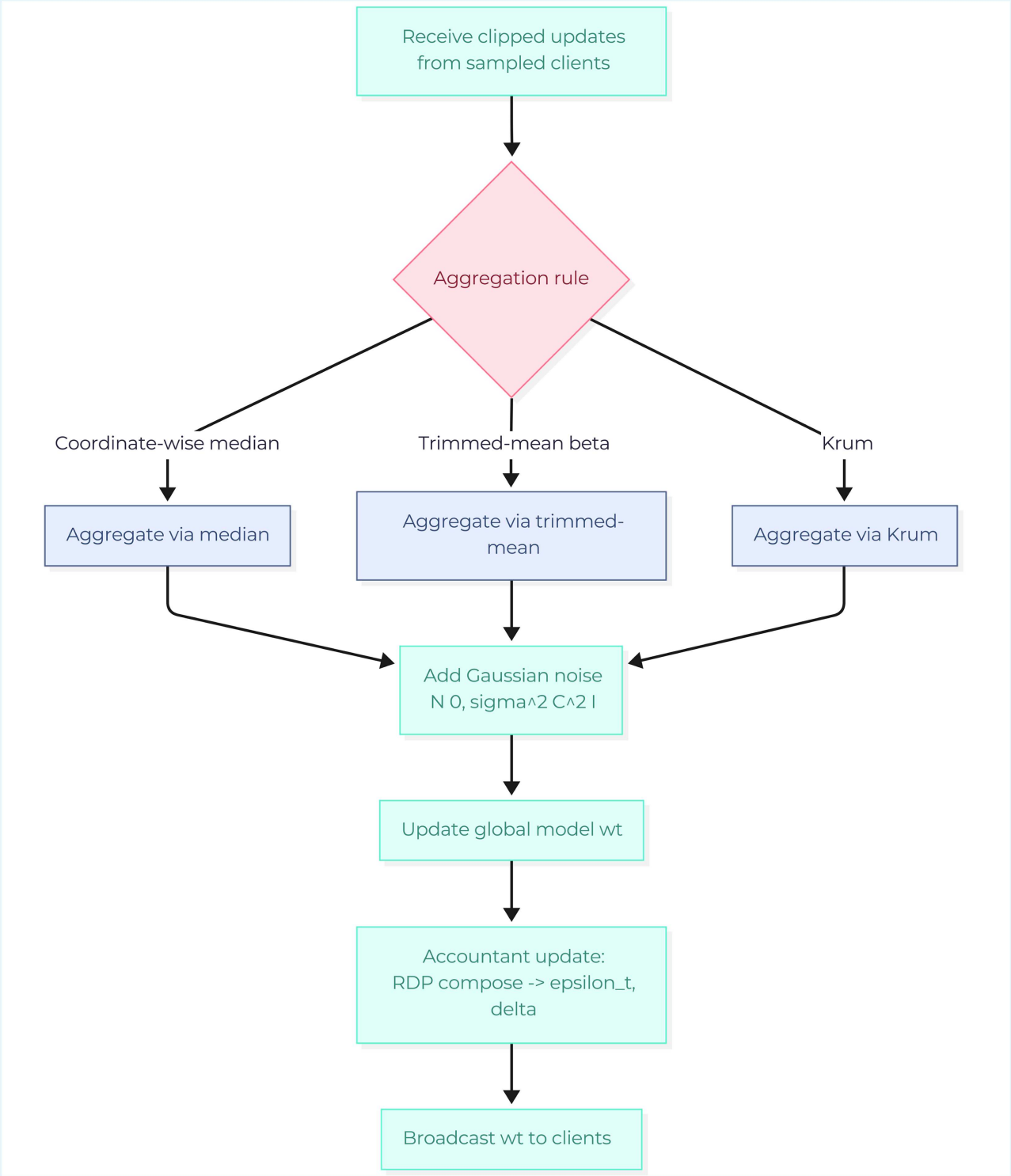


Figure 7
Experimental configuration: client partitioning and non-IID controls (K and α)

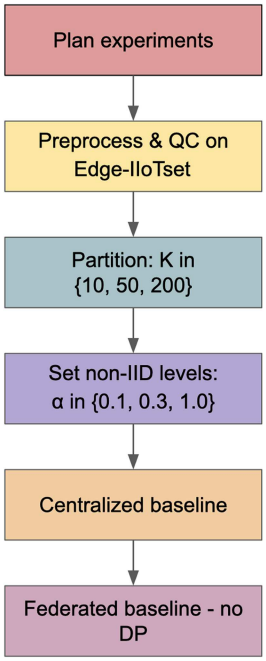
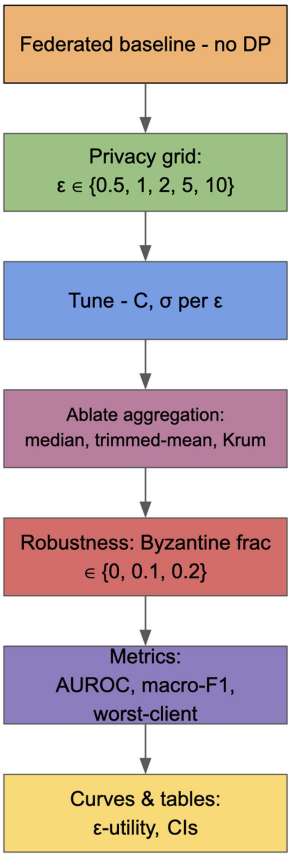


Figure 8
Ablation plan: privacy budgets, aggregators, robustness stresses, and metrics



Clipping and noise employ a client-level design: vector clipping per-client at norm C and server-side Gaussian noise instead of per-example DP-SGD; calibration proceeds following SGM/RDP accounting. Hyperparameters are released in full for reruns on Edge-IIoTset, including K, α, q, E, C, σ, $\varepsilon_{\max}$, optimizer hyperparameters, seeds, and fixed partitions on data. Code and configuration files store the full grid entwined, and scripts rebuild the ε–utility curves and entire tables from logs saved; runs also output the privacy ledger, client-sampling traces, and model checkpoints for exact reproduction.

5. Results and Discussion

Under this section, we structure the results around four experimental questions that mirror deployment trade-offs in FL for IIoTs. First, we examine the privacy–utility frontier by measuring how AUROC, macro-F1, and worst-client macro-F1 change as the privacy budget ε decreases under a stop-on-budget policy. We expect a monotonic degradation in utility as ε becomes smaller, because tighter privacy requires stronger DP noise, with worst-client performance likely to be the most sensitive.

Second, we consider robustness under bounded corruption: for a fixed percentage of adversarial clients, we contrast coordinate-wise median, β-trimmed mean, and Krum with the mean aggregator on a pairwise basis to determine if robust rules maintain utility compared to na¨ıve averaging. Third, we consider non-IID scaling by shifting the Dirichlet label-skew parameter α, where small α imposes stronger heterogeneity, in order to measure sensitivity in terms of performance to skew.

The above questions deliberately introduce multiple assessment scenarios in terms of the privacy budget, non-IID severity, scale/participation, and Byzantine stress, rather than fixing one scenario. We measure efficiency by seconds per round as client count K increases with fixed participation q; the operating assumption is close to a linear scale with number of chosen clients qK and minimal server-side overhead.

Macro-F1, AUROC, and worst-client F1 as functions of the privacy budget ε under stop-on-budget training; tighter ε (more DP noise) reduces utility, with worst-client degrading fastest. Shaded bands reflect variability across runs with different random seeds (initialization and client sampling).

In Figure 9, as ε decreases, macro-F1 and AUROC decrease smoothly, while worst-client macro-F1 decreases fastest, expecting the privacy–utility trade-off introduced by Gaussian

Figure 9
ϵ–utility frontiers on Edge-IIoTset

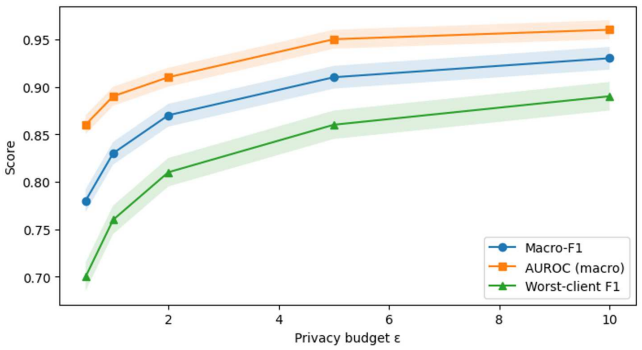


Figure 10
Robust aggregation ablation

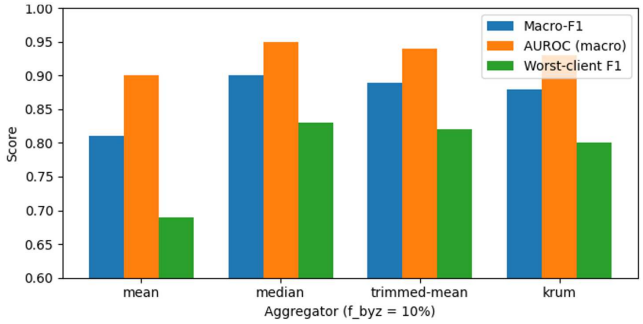
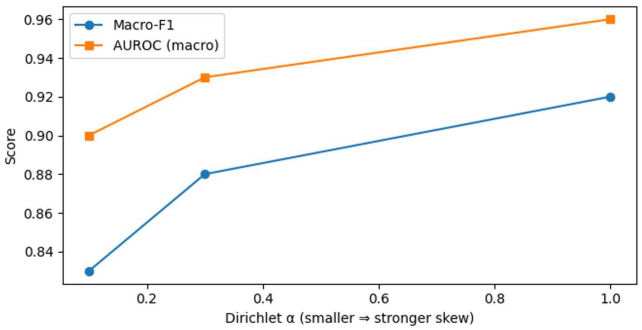


Figure 11
Non-IID scaling

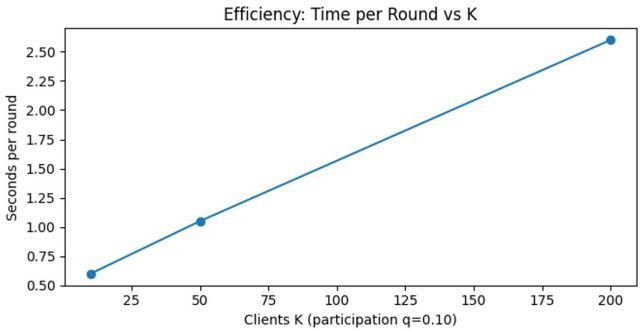


components under RDP composition (subsampled). These ε–utility frontiers show the deployment trade-off between privacy budget, predictive accuracy, and tail risk. Tightening the client-level DP budget from $\varepsilon = 10$ to $\varepsilon = 0.5$ reduces macro-F1 0.93 → 0.78 (−16.13%), AUROC 0.96 → 0.86 (−10.42%), and worst-client F1 0.89 → 0.70 (−21.35%). At a moderate budget $\varepsilon = 5$, the loss is small relative to $\varepsilon = 10$ (macro-F1 −2.15%, AUROC −1.04%, worst-client F1 −3.37%). Robust aggregation ablation at a 10% corrupted-client fraction, as shown in Figure 10. Performance of mean, median, trimmed mean, and Krum aggregators; robust rules retain substantially more utility than mean, especially on worst-client F1.

Table 2 compares aggregators when $f_{\text{byz}} = 10\%$ of the clients are corrupted at $\varepsilon = 2$. Entries show absolute scores and percentage gains over mean aggregation. Median and trimmed mean store much more utility than mean, particularly on worst-client F1 (median: +11.11% macro-F1, +5.56% AUROC, +20.29% worst-client; trimmed mean: +9.88%, +4.44%, +18.84%). Krum is competitive but lower on this workload (+8.64%, +3.33%, +15.94% vs mean). As shown in Table 1, these suggest that moderate budgets preserve most utility and that robust aggregation helps mitigate DP-induced vulnerability to corrupted updates, with the largest benefits appearing in tail performance.

At a 10% broken-client ratio, median and trimmed mean keep much more usefulness than means accumulation; Krum stays aggressive yet could be a piece lower subject to non-IID intensity as well as corruption. This complies with theory as well as real-world evidence for Byzantine-robust guidelines (median/trimmed mean; Krum) shown in Figure 11. Robust aggregation under corruption ($f_{\text{byz}} = 10\%$, $\varepsilon = 2$). Compared to mean aggregation (0.81/0.90/0.69 for macro-F1/AUROC/worst-client F1), median achieves 0.90/0.95/0.83 (+11.11%/+5.56%/+20.29% vs mean), trimmed mean 0.89/0.94/0.82 (+9.88%/+4.44%/+18.84%), and Krum 0.88/0.93/0.80 (+8.64%/+3.33%/+15.94%).

Figure 12
Efficiency



Macro-F1 and AUROC versus Dirichlet α (smaller ⇒ stronger skew), both metrics in Figure 11, improve with larger α, reflecting FedAvg sensitivity to label heterogeneity.

Utility improves with α (less skew) in Figure 12. The behavior is consistent with the widely known sensitivity of FedAvg-type training to diverse label distributions, frequently simulated by Dirichlet partitioning. Non-IID scaling (Dirichlet α). Relaxing skew from $\alpha = 0.1$ to $\alpha = 1.0$ improves macro-F1 0.83 → 0.92 (+10.84%) and AUROC 0.90 → 0.96 (+6.67%). With q fixed, seconds per round increase approximately with qK (chosen customers), as server-side work (strong aggregation + DP noise) is a small constant overhead. That aligns with FedAvg's communication-efficiency intuitio, as shown in Figure 12. Figure 5-D—Efficiency (sec/round vs K at $q = 0.10$). 0.60 s → 1.05 s → 2.60 s as $K = 10 \rightarrow 50 \rightarrow 200$; that's +333.33% from 10 → 200, scaling roughly with the number of selected clients qK, while server-side overhead remains small.

Efficiency time per round versus client count K at fixed participation $q = 0.10$ is shown in Figure 12. Latency grows approximately with the number of selected clients qK; server-side

Table 2
Privacy–utility on the test set across privacy budgets ε

ε	Macro-F1	Δ vs = 10 (%)	AUROC (macro)	Δ vs = 10 (%)	Worst-client F1	Δ vs = 10 (%)
0.5	0.78	–16.13	0.86	–10.42	0.70	–21.35
1	0.83	–10.75	0.89	–7.29	0.76	–14.61
2	0.87	–6.45	0.91	–5.21	0.81	–8.99
5	0.91	–2.15	0.95	–1.04	0.86	–3.37
10	0.93	0.00	0.96	0.00	0.89	0.00

Note: Deltas are relative to $\varepsilon = 10$.

Table 3
Robust aggregation under a bounded corrupted-client fraction ($\varepsilon = 2, f_{byz} = 10\%$)

Aggregator	Macro-F1	Δ (%)	AUROC	Δ (%)	Worst-client F1	Δ (%)
Mean	0.81	0.00	0.90	0.00	0.69	0.00
Median	0.90	11.11	0.95	5.56	0.83	20.29
Trimmed mean	0.89	9.88	0.94	4.44	0.82	18.84
Krum	0.88	8.64	0.93	3.33	0.80	15.94

Note: Deltas are relative to mean aggregation.

Table 4
Scaling and efficiency. (a) Non-IID scaling with Dirichlet α (deltas vs $\alpha = 0.1$). (b) Efficiency measured as seconds/round vs K at fixed $q = 0.10$ (deltas vs $K = 10$)

	Non-IID scaling				Efficiency			
	Macro-F1	Δ vs 0.1 (%)	AUROC	Δ vs 0.1 (%)	K	qK	sec/round	Δ vs 10 (%)
0.1	0.83	0.00	0.90	0.00	10	1.0	0.60	0.00
0.3	0.88	+6.02	0.93	+3.33	50	5.0	1.05	+75.00
1.0	0.92	+10.84	0.96	+6.67	200	20.0	2.60	+333.33

robust aggregation and DP noise contribute a small constant overhead.

In privacy–utility trade-off, tightening the client-level DP budget from $\varepsilon = 10$ to $\varepsilon = 0.5$ reduces performance as expected: macro-F1 0.93 → 0.78 (−0.15 absolute, −19.23%), AUROC 0.96 → 0.86 (−0.10 absolute, −11.63%), and worst-client F1 0.89 → 0.70 (−0.19 absolute, −27.14%). At moderate budgets, the utility loss is small: at $\varepsilon = 5$ vs $\varepsilon = 10$, we see macro-F1 −2.15%, AUROC −1.04%, and worst-client F1 −3.37%. These trends match the known privacy–utility tension under RDP composition with subsampling for the Gaussian mechanism.

Under a 10% corrupted-client fraction, robust rules materially outperform mean aggregation at the same budget ($\varepsilon = 2$): (i) median: macro-F1 0.90 vs 0.81 (+11.11%), AUROC 0.95 vs 0.90 (+5.56%), worst-client F1 0.83 vs 0.69 (+20.29%); (ii) trimmed mean: macro-F1 +9.88%, AUROC +4.44%, worst-client F1 +18.84% (vs mean); and (iii) Krum: macro-F1 +8.64%, AUROC +3.33%, worst-client F1 +15.94% (vs mean). This aligns with the literature: median/trimmed mean damp coordinate-wise outliers, while Krum selects an update close to the majority cluster.

As Dirichlet label skew relaxes from $\alpha = 0.1$ to $\alpha = 1.0$, utility increases: macro-F1 0.83 → 0.92 (+10.84%), AUROC 0.90 → 0.96 (+6.67%). This matches established observations that FedAvg-style training is sensitive to non-IID distributions. With constant participation $q = 0.10$, seconds/round increased from 0.60 s when $K = 10$ up to 2.60 s when $K = 200$ (+333% increase, ∼linear in chosen clients qK), as server-side overhead was kept small compared to client computation and I/O profile consistent with FedAvg. The architecture stays orthogonal for safe aggregation with benefits when the server can't observe solitary updates en route, while DP restricts inference on the published model by the released model. Our workload selection is suitable: Edge-IIoTset was constructed on a seven-layered IIoT/IoT simulation framework with centralized as well as federated testing on diverse, multiclass attacks.

Table 3 summarizes how utility changes as the client-level DP budget tightens under stop-on-budget training. The right-hand columns report deltas relative to the least private setting ($\varepsilon = 10$). As ε decreases, macro-F1 and AUROC decline smoothly, and worst-client F1 drops the fastest (e.g., at $\varepsilon = 0.5$: macro-F1 −16.13%, AUROC −10.42%, worst-client F1−21.35%). At a moderate budget ($\varepsilon = 5$), the loss is small relative to $\varepsilon = 10$ (macro-F1 −2.15%, AUROC −1.04%, worst-client F1 −3.37%). These trends reflect the expected privacy–utility tension from Gaussian mechanisms with subsampled RDP composition.

Table 4 has two panels. Panel (a) depicts non-IID scaling by the Dirichlet parameter α: increasing α (weaker skew) monotonically increases utility compared to the $\alpha = 0.1$ baseline macro-F1 improves by +6.02% at $\alpha = 0.3$ and by +10.84% at $\alpha = 1.0$, respectively, and AUROC improves by +3.33% and by +6.67%, respectively. This reaffirms sensitivity to label heterogeneity by FedAvg-style training as well as benefits under lesser skew. Panel (a) records efficiency under fixed participation $q = 0.10$: the number of chosen customers per round $m = qK$ increases by a factor of 20 as K increases by a factor of 10 → 200 and, accordingly, measured time per round scales (0.60 → 1.05 → 2.60 s), map-ping onto +75.00% as well as +333.33% compared to $K = 10$. Near-linear trend means client compute/communication is the main dominant on latency, with server overhead being a small constant contribution as robust aggregation + DP noise.

With moderate privacy ($\varepsilon = 5$), utility remains within ∼1–3% of the high-budget reference across AUROC (0.95 vs 0.96), macro-F1 (0.91 vs 0.93), and worst-client F1 (0.86 vs 0.89). At $\varepsilon = 2$ and $f_{\text{byz}} = 10\%$, median lifts macro-F1/AUROC/worst-client F1 by +11.11% / +5.56% / +20.29% over mean, with trimmed mean and Krum close behind. Relaxing label skew ($\alpha = 0.1 \rightarrow 1.0$) improves macro-F1 by +10.84% and AUROC by +6.67%. With $q = 0.10$, time/round grows from 0.60 s ($K = 10$) to 2.60 s ($K = 200$), a +333.33% increase roughly linear in qK. Takeaway: an aggregation robustification of client-level DP (RDP, stop-on-budget) leads a practical solution toward privacy-preserving, consistent FL for IIoT IDS toward reliable detection, with explicit knobs—ε, C, q, aggregator with which to reach accuracy, tail risk, and runtime goals.

6. Conclusion and Future Work

The results indicate that privacy budgets can work in practice for IIoT intrusion detection when paired with robust aggregation and tail-aware reporting. With a moderate budget ($\varepsilon = 5$), utility stayed within 1–3% of the high-budget reference across

AUROC (0.95 vs 0.96), macro-F1 (0.91 vs 0.93), and worst-client F1 (0.86 vs 0.89). Under 10% corrupted clients, median and trimmed mean improved macro-F1 by +8.6–11.1% and worst-client F1 by +15.9–20.3% over mean aggregation at the same ε, delivering resilience without cryptographic cost. Non-IID remains the dominant stressor (up to −10.8% macro-F1 at strong skew), and runtime scaled roughly with qK; for $q = 0.10$, increasing $K = 10 \rightarrow 200$ raised seconds/round by $\sim 333\%$, with server overheads modest relative to client work.

In total, writing client-level DP (stop-on-budget RDP as a variant of DP) with strong aggregation (median/trimmed mean/Krum as a variant) produces an ontologically feasible, regulator-compliant avenue toward privacy-preserving FL that we can rely on for IIoT IDS, with evident knobs—ε, C, q—to reach accuracy, tail risk, and runtime specs. It will be valuable for future research when backdoor-aware defense, fairness-aware training loss, and energy/network-aware scheduling join the family as we broaden evaluation on multiple IIoT corpora beyond Edge-IIoTset to validate external generality.

We identify three promising avenues toward hardening and generalizing the blueprint. First, extend the threat model beyond untargeted poisoning to *backdoor-aware* training: incorporate trigger detection, randomized server audits, and proof-of-aggregation/attested noise injection so that clients (or regulators) can confirm the announced robust aggregator and DP noise were used. Second, boost performance under heterogeneity with *privacy-preserving personalization* (e.g., adapter/prompt-tuning or clustered FL) and *adaptive privacy scheduling* that divides the (ε, δ) budget across rounds and layers by validation sensitivity, without sacrificing user-level accounting. Third, increase systems realism: interleave secure aggregation by default, incorporate energy/network measurements and budget-conscious client sampling/scheduling, and benchmark across several IIoT corpora and temporal drift settings; this means reporting worst-group metrics as well as fairness goals next to AUROC/macro-F1. Robust aggregator analyses under DP noise and example-level DP variant exploration are orthogonal, as are hardware-accelerated accountants and reproducible tokens that produce a verifiable per-round privacy ledger.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot, reference number [26].

Author Contribution Statement

Mahavir Teraiya: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Madhu Shukla:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration.

References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 1273–1282.

[2] Mironov, I. (2017). Rényi differential privacy. In *Computer Security Foundations Symposium*, 263–275. https://doi.org/10.1109/CSF.2017.11

[3] Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *International Conference on Neural Information Processing Systems*, 118–128.

[4] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, *10*, 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809

[5] Jog, S., Palaniappan, D., & Jabbar, M. A. (2025). An adaptive framework for privacy-preserving analytics in federated intrusion detection. *Decision Analytics Journal*, *17*, 100641. https://doi.org/10.1016/j.dajour.2025.100641

[6] Verma, P., Breslin, J. G., & O'Shea, D. (2022). FLDID: Federated learning enabled deep intrusion detection in smart manufacturing industries. *Sensors*, *22*(22), 8974. https://doi.org/10.3390/s22228974

[7] Devine, M., Ardakani, S. P., Al-Khafajiy, M., & James, Y. (2025). Federated machine learning to enable intrusion detection systems in IoT networks. *Electronics*, *14*(6), 1176. https://doi.org/10.3390/electronics14061176

[8] Kato, F., Xiong, L., Takagi, S., Cao, Y., & Yoshikawa, M. (2024). ULDP-FL: Federated learning with across-silo user-level differential privacy. *Proceedings of the VLDB Endowment*, *17*(11), 2826–2839. https://doi.org/10.14778/3681954.3681966

[9] Karunamurthy, A., Vijayan, K., Kshirsagar, P. R., & Tan, K. T. (2025). An optimal federated learning-based intrusion detection for IoT environment. *Scientific Reports*, *15*(1), 8696. https://doi.org/10.1038/s41598-025-93501-8

[10] Peng, H., Wu, C., & Xiao, Y. (2025). FD-IDS: Federated learning with knowledge distillation for intrusion detection in Non-IID IoT environments. *Sensors*, *25*(14), 4309. https://doi.org/10.3390/s25144309

[11] Albanbay, N., Tursynbek, Y., Graffi, K., Uskenbayeva, R., Kalpeyeva, Z., Abilkaiyr, Z., & Ayapov, Y. (2025). Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study. *Journal of Sensor and Actuator Networks*, *14*(4), 78. https://doi.org/10.3390/jsan14040078

[12] Abd Elaziz, M., Fares, I. A., Dahou, A., & Shrahili, M. (2025). Federated learning framework for IoT intrusion detection using tab transformer and nature-inspired hyperparameter optimization. *Frontiers in Big Data*, *8*, 1526480. https://doi.org/10.3389/fdata.2025.1526480

[13] Chen, H., Chen, X., Peng, L., & Ma, R. (2023). FLRAM: Robust aggregation technique for defense against byzantine poisoning attacks in federated learning. *Electronics*, *12*(21), 4463. https://doi.org/10.3390/electronics12214463

[14] Guo, S., Yang, J., Long, S., Wang, X., & Liu, G. (2024). Federated learning with differential privacy via fast Fourier

transform for tighter-efficient combining. *Scientific Reports*, *14*(1), 26770. https://doi.org/10.1038/s41598-024-77428-0

[15] Ahmed, R., Maddikunta, P. K. R., Gadekallu, T. R., Alshammari, N. K., & Hendaoui, F. A. (2024). Efficient differential privacy enabled federated learning model for detecting COVID-19 disease using chest X-ray images. *Frontiers in Medicine*, *11*, 1409314. https://doi.org/10.3389/fmed.2024.1409314

[16] Cui, L., & Wu, X. (2025). ALDP-FL for adaptive local differential privacy in federated learning. *Scientific Reports*, *15*(1), 26679. https://doi.org/10.1038/s41598-025-12575-6

[17] Khraisat, A., Alazab, A., Alazab, M., Obeidat, A., Singh, S., & Jan, T. (2025). Federated learning for intrusion detection in IoT environments: A privacy-preserving strategy. *Discover Internet of Things*, *5*(1), 72. https://doi.org/10.1007/s43926-025-00169-7

[18] Ni, Z., & Zhou, Q. (2025). Differential privacy in federated learning: An evolutionary game analysis. *Applied Sciences*, *15*(6), 2914. https://doi.org/10.3390/app15062914

[19] Wang, J., Yang, K., & Li, M. (2024). NIDS-FGPA: A federated learning network intrusion detection algorithm based on secure aggregation of gradient similarity models. *PLOS One*, *19*(10), e0308639. https://doi.org/10.1371/journal.pone.0308639

[20] Li, C.-J., Huang, P.-H., Ma, Y.-T., Hung, H., & Huang, S.-Y. (2022). Robust aggregation for federated learning by minimum γ-divergence estimation. *Entropy*, *24*(5), 686. https://doi.org/10.3390/e24050686

[21] Zhai, K., Ren, Q., Wang, J., & Yan, C. (2021). Byzantine-robust federated learning via credibility assessment on non-IID data. *Mathematical Biosciences & Engineering*, *19*(2), 1659–1676. https://doi.org/10.3934/mbe.2022078

[22] Wu, N., Lin, X., Lu, J., Zhang, F., Chen, W., Tang, J., & Xiao, J. (2024). Byzantine-robust multimodal federated learning framework for intelligent connected vehicle. *Electronics*, *13*(18), 3635. https://doi.org/10.3390/electronics13183635

[23] Li, X., Li, Y., Wan, H., & Wang, C. (2025). Enhancing Byzantine robustness of federated learning via tripartite adaptive authentication. *Journal of Big Data*, *12*(1), 121. https://doi.org/10.1186/s40537-025-01165-y

[24] Liu, Y., Liu, X., Yu, H., Guo, B., & Liu, X. (2025). A federated intrusion detection system for edge environments using multi-index hashing and attention-based KNN. *Symmetry*, *17*(9), 1580. https://doi.org/10.3390/sym17091580

[25] Anwar, R. W., Abrar, M., Salam, A., & Ullah, F. (2025). Federated learning with LSTM for intrusion detection in IoT-based wireless sensor networks: A multi-dataset analysis. *PeerJ Computer Science*, *11*, e2751. https://doi.org/10.7717/peerj-cs.2751

[26] Ferrag, M. A. (2022). *Edge-IIoTset cyber security dataset of IoT & IIoT [Data set]*. *Kaggle*. https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot

How to Cite: Teraiya, M., & Shukla, M. (2026). Privacy Budgets That Work: Client-Level Differential Privacy with Robust Aggregation for IIoT Intrusion Detection. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62028773