OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 12 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

CALM-FEDIDS: ROBUST FEDERATED INTRUSION DETECTION ON HETEROGENEOUS DATA

Wahaj Hassan Soomro[1], Muhammad Arslan Siddiqui[2]

[1]*Institute of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan.*
[2]*FAST NUCES, Karachi, Sindh, Pakistan.*

[*]***Corresponding Author:*** (wahajhassansoomro@gmail.com)
DOI:(*https://doi.org/10.71146/kjmr795*)

Article Info

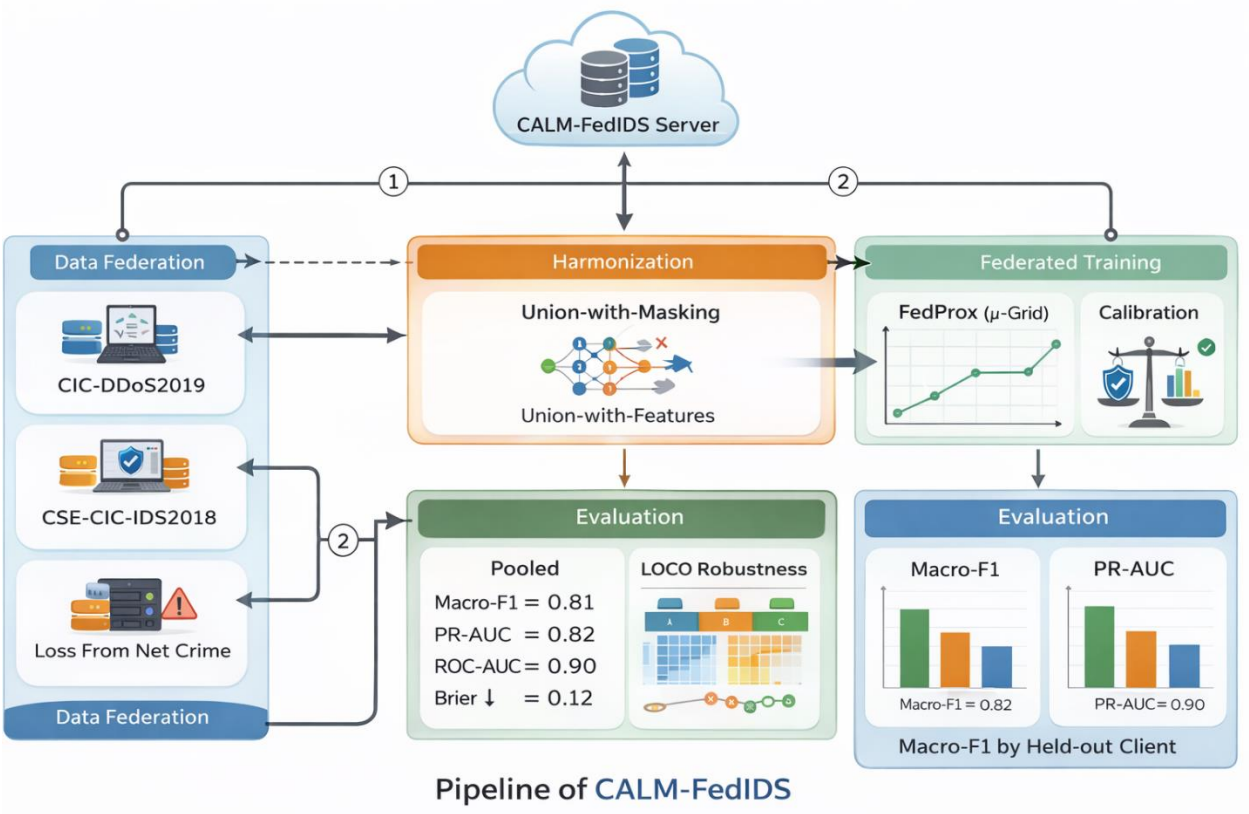
Abstract

Network attack detectors often fail when data come from different places, follow different schemas, or cannot be centralized for privacy reasons. We study this practical setting using five real datasets (CIC-DDoS2019, CSE-CIC-IDS2018, Friday-Working Hours-Afternoon-DDoS, Loss-From-Net-Crime, and CyS_Attacks_Dataset) that differ in feature names, class balance, and file formats. To the best of our knowledge, prior IDS work has not rigorously trained and evaluated on truly heterogeneous, multi-source datasets without centralizing raw data. Our goal is a deployable detector that (i) learns without pooling raw data, (ii) remains robust when a site has new or missing features, and (iii) produces calibrated probabilities that translate directly into risk thresholds. We propose CALM-FedIDS—a *Calibrated, Alignment-and-Masking* federated method built on FedProx. First, we harmonize features by name and create a union-with-masking representation. We train a lightweight logistic model with FedProx (μ-grid search) so sites with non-IID data can still contribute safely. We evaluate client hold-out (LOCO) to mimic deployment at an unseen site, and we report class-imbalance-aware metrics (macro-F1, PR-AUC) and calibration (reliability curves, Brier). We also compare against a centralized oracle and per-client baselines. On pooled evaluation, CALM-FedIDS achieves strong discrimination (ROC-AUC ≈ 0.913, PR-AUC ≈ 0.829) and good probability quality, outperforming a centralized oracle baseline (ROC-AUC ≈ 0.828, PR-AUC ≈ 0.672) while keeping data local. CALM-FedIDS delivers a practical IDS for multi-institution settings: it respects privacy, tolerates schema drift, and outputs calibrated scores that operations teams can threshold. For deployment, (1) keep the union-with-masking schema log across partners, (2) select μ using a small validation exchange of *metrics only*, (3) monitor calibration and update post-hoc (e.g., isotonic) if traffic shifts, and (4) extend with secure aggregation or differential privacy when policy requires stronger guarantees.

Keywords: *Federated Learning, FedProx, Intrusion Detection System, Heterogeneity, Security*

Introduction

Modern intrusion detection systems (IDS) are trained on data that are siloed across institutions, shaped by non-IID traffic patterns, and constrained by privacy and governance rules that prevent centralizing raw packets or flow logs. Even when data can be shared, feature schemas differ across sites (names, units, and availability), which quietly breaks many published pipelines that assume aligned columns or a single dataset [1]– [3]. As a result, models that look strong in-distribution often falter when deployed at a new organization with different sensors or missing fields—and the community rarely reports calibration or cross-site robustness even though security operations depend on *threshold able, well-calibrated risk scores* rather than raw logits [4], [5]. We target a deployable IDS that (i) learns without pooling raw data (federated training), (ii) tolerates schema drift and partial features across sites, and (iii) emits calibrated probabilities that translate into actionable thresholds.



End-to-end pipeline of CALM-FedIDS. Heterogeneous intrusion detection datasets remain local at each client and are harmonized using a union-with-masking schema. A lightweight model is trained via federated optimization (FedProx with μ-grid selection) and calibrated for reliable probability estimates. Performance is evaluated using pooled metrics and leave-one-client-out (LOCO) robustness to assess cross-institution generalization.

We introduce CALM-FedIDS—*Calibrated, Alignment-and-Masking Federated IDS*. First, we harmonize columns by name and build a union-with-masking representation: every feature seen anywhere is included; missing features at a site are zero-filled, and paired mask indicators record absence (so the model can learn from "missingness" rather than be misled by imputation). Each client standardizes locally. Second, we train a lightweight logistic model with FedAvg/FedProx to

handle heterogeneous clients [6], [7]. Third, we evaluate leave-one-client-out (LOCO) to mimic deployment at an unseen site (a standard out-of-distribution test conceptually aligned with WILDS-style evaluation [8], [9]). Finally, we report class-imbalance-aware metrics (macro-F1, PR-AUC) and calibration (Brier score, reliability curves) because these determine deploy ability in skewed attack settings [4], [10].

Contributions.

1. Schema harmonization for FL-IDS. A simple, reproducible union-with-masking that preserves local schemas and exposes missingness to the model—no global feature intersection required.
2. Federated robustness protocol. A LOCO evaluation across three public IDS families (CIC-DDoS2019, CSE-CIC-IDS2018, and CIC-IDS2017 Friday-Working Hours) that stresses generalization to unseen sites/days while keeping data local [1]– [3].
3. Calibration-first reporting. We pair discrimination (ROC/PR) with reliability (Brier + reliability diagrams) and show how post-hoc calibration (e.g., Platt or isotonic) improves threshold ability [4], [11].
4. Practical baselines. We situate CALM-FedIDS between centralized oracle and per-client local baselines, and include a small μ-grid ablation (FedAvg vs FedProx) and union vs intersection features.

High-level results (brief teaser). Across the three core datasets, CALM-FedIDS yields pooled macro-F1 around ~0.8, PR-AUC competitive for imbalanced clients, and well-behaved reliability curves. LOCO shows non-trivial transfer to held-out clients without collapsing the minority class. These results support CALM-FedIDS as a practical baseline for FL-IDS: privacy-respecting, schema-tolerant, and calibration-aware—not a SOTA deep ensemble, but a transparent, deployable approach that others can extend with secure aggregation or DP when required [12], [13].

Related Work

Classic network IDS pipelines often assume centralized access to full traffic or flow and aligned features, reporting high in-distribution accuracy on single datasets (e.g., CIC-IDS2017, CSE-CIC-IDS2018, CIC-DDoS2019) [1]– [3]. However, cross-dataset generalization is weak if schemas differ or traffic shifts; recent analyses highlight the drop when moving across sources or time [14]. Our LOCO protocol addresses this by holding out an entire site/day during training.

FL has been proposed to respect data locality while aggregating model updates [6]. FedProx stabilizes optimization under system and statistical heterogeneity via a proximal term [7]. Prior FL-IDS works often report accuracy/AUC on a single dataset or on curated splits, with limited reporting of calibration or held-out-site robustness (surveys note these gaps) [5]. We adopt federated logistic regression as a transparent baseline, run a μ-grid (FedAvg→FedProx), and evaluate LOCO to make transfer explicit.

Heterogeneous tabular sensors produce non-matching schemas; naive intersection discards information, while uninformed imputation can bias models. A well-established trick in tabular/time-series is to include missingness indicators so models can learn that “feature X absent”

carries signal [15], [16]. Our union-with-masking operationalizes this at scale for FL-IDS and avoids brittle, global schema coordination.

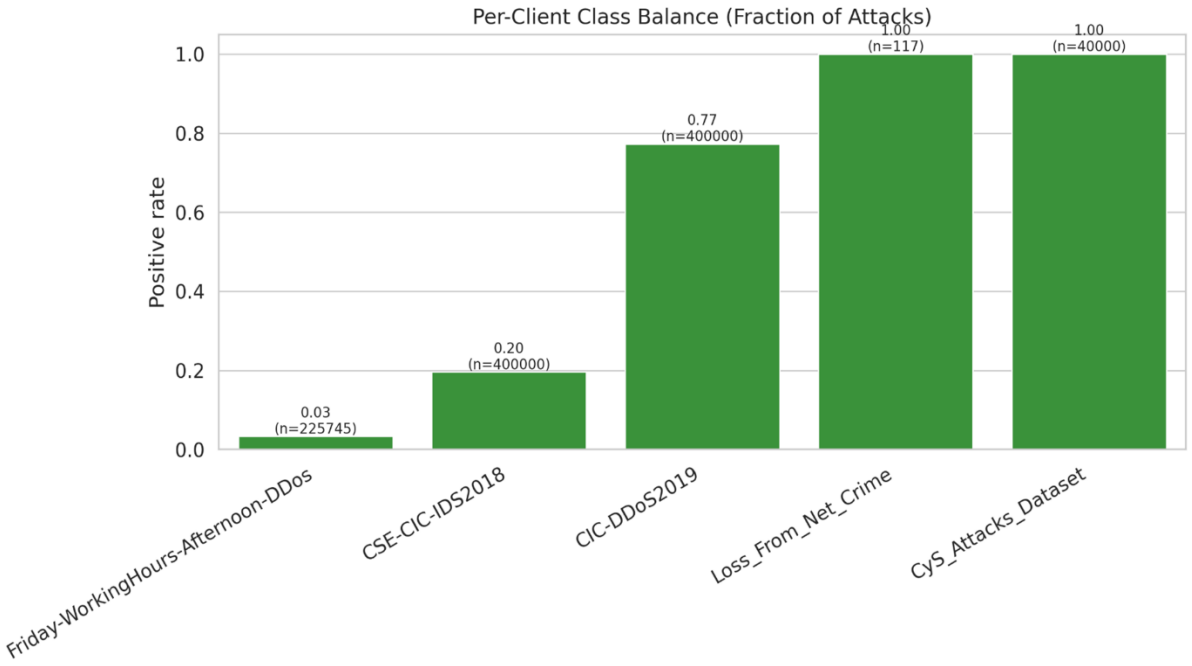
In skewed attack detection, PR-AUC is more informative than ROC-AUC [10]. Meanwhile, mis-calibration yields unusable thresholds in operations. Guo et al. showed that modern models are often over-confident; Brier and reliability curves expose this, and Platt/isotonic refine probabilities [4], [11]. We make calibration a first-class metric and include post-hoc calibration in our reporting.

Evaluating on unseen groups (hospital/site/day) captures real deployment risk. The WILDS line of work formalizes such distribution-shift evaluation [8], [9]. Our LOCO design follows this spirit for IDS: train on {A, B…}, test on held-out C to reflect installation at a new organization.

While our focus is methodological evaluation, secure aggregation and differential privacy can harden FL pipelines [12], [13]. These are drop-in extensions to CALM-FedIDS when policies demand stronger guarantees.

Data Card

This work uses five real-world IDS sources that differ in file formats, feature names, sampling windows, and label practices. Rather than force a single, perfect schema, we treat each source as a client in a federated setting and document its quirks up front. Below, "positive" means *attack/malicious* and "negative" means *benign/normal*.



Per-client attack prevalence; severe skew motivates PR-AUC and calibration.

Common label semantics (how we build y)

Explicit labels: If a dataset ships a `Label/Attack/Class` field, we map `{Benign, Normal}`→0 and all attack strings (e.g., `DDoS`, `DrDoS_DNS`, `SYN`, `UDP`, `ICMP`, `Botnet`)→1.
Numeric flags: If the label is numeric, `0`→0 and any `>0`→1.
Derived labels (when no explicit label):
Loss_From_Net_Crime: We derive `y=1` if a numeric loss/amount/cost/damage field is `> 0`, else `0`.
CyS_Attacks_Dataset: We derive from a categorical field like `attack_type/category/type`, mapping `{benign, normal}`→0 and all others→1.

We keep this mapping consistent across clients so pooled and LOCO metrics are comparable.

Client 1 — CIC-DDoS2019

What it is: Modern DDoS traffic with multiple reflection/amplification types; typically distributed as many Parquet/CSV shards with flow features.
Schema notes: Rich flow statistics (forward/backward packet lengths, inter-arrival times, flags); names vary slightly across shards.
Class balance: Skewed toward attacks (positive rate ≈ 0.77 in our sample).
Takeaways for modeling: High positive rate helps AUC/accuracy but can hide false positives; PR-AUC is the fairer score. Good stress-test for *calibration* because over-confident scores will look deceptively "good" under ROC alone.

Client 2 — CSE-CIC-IDS2018

What it is: Multi-day, multi-attack benchmark with diverse traffic; often the backbone of cross-paper comparisons.
Schema notes: Overlaps with CIC-style features but not identical; column names and units may vary; some features missing per split.
Class balance: Moderately imbalanced with fewer positives (positive rate ≈ 0.20 in our sample).
Takeaways for modeling: Strong test for recall on minority attacks and for LOCO transfer when held out; masks matter because not all columns appear on every day.

Client 3 — Friday-Working Hours (CIC-IDS2017 slice)

What it is: The well-known "Friday" subset (Working Hours) from CIC-IDS2017; widely used, sometimes too clean compared to production.
Schema notes: Overlaps conceptually with CIC-style features; still, some renaming's and missing columns occur.
Class balance: Highly imbalanced toward benign (positive rate ≈ 0.03).
Takeaways for modeling: This client pressures precision–recall; models that look great on ROC can still flood operators with false alerts here unless probabilities are well calibrated.

Client 4 — Loss_From_Net_Crime *(derived labels)*

What it is: A small, tabular summary of cyber events with monetary loss fields.
Schema notes: Few columns; no explicit "attack" label. We derive `y=1` when loss/amount/cost/damage > 0.
Class balance: All positive in our pull (positive rate ≈ 1.00).
Takeaways for modeling: Cannot be used for training a binary classifier (no negatives), but it is informative for evaluation (probability ranking), calibration curves, or cost-aware post-processing. In federated training, we exclude it from gradient aggregation (to avoid collapse) but still report evaluation where appropriate.

Client 5 — CyS_Attacks_Dataset *(derived labels)*

What it is: A compact catalog of attack classes curated for teaching/analysis.
Schema notes: Fewer features; categorical attack column is the main signal. We map `{benign, normal}` →0, others→1.
Class balance: All positive in our pull (positive rate ≈ 1.00).
Takeaways for modeling: Same as above—use carefully. It is helpful for out-of-domain sanity checks and confusion-matrix illustrations, but not for federated weight updates unless a non-attack slice becomes available.

Per-client skews and why they matter

Imbalance spans both extremes: Friday-Working Hours is benign-heavy; CIC-DDoS2019 is attack-heavy. This is why we report macro-F1 and PR-AUC, not just accuracy/ROC.

Schema misalignment is the rule: Feature names, units, and availability differ. Our union-with-masking keeps every column seen anywhere, fills missing values with zeros, and adds a mask bit per feature so the model can learn from "present vs. absent."

All-positive clients exist in practice: Organizational summaries or curated attack catalogs often lack benign rows. We exclude them from federated training (to avoid degenerate gradients) but still use them as evaluation clients where appropriate (e.g., ranking, qualitative transfer, or cost curves).

Pre-processing summary (what we actually do)

Column harmonization: Normalize names (case, punctuation, common synonyms like `fwd/backward`, `src/dst`) and build a global union of features.
Local standardization: Each client standardizes *its present* columns (z-score using that client's stats).
Masking: For each feature in the union, create `feature` (value or 0) and `feature__mask` (1 if present, 0 if absent).
Label coercion: Apply the consistent mapping above; for derived cases, use numeric loss or attack category rules.
Splits: Within each client we keep a small validation slice; for LOCO, we hold out an entire client at test time to mimic a new site.

How to read our metrics against this card

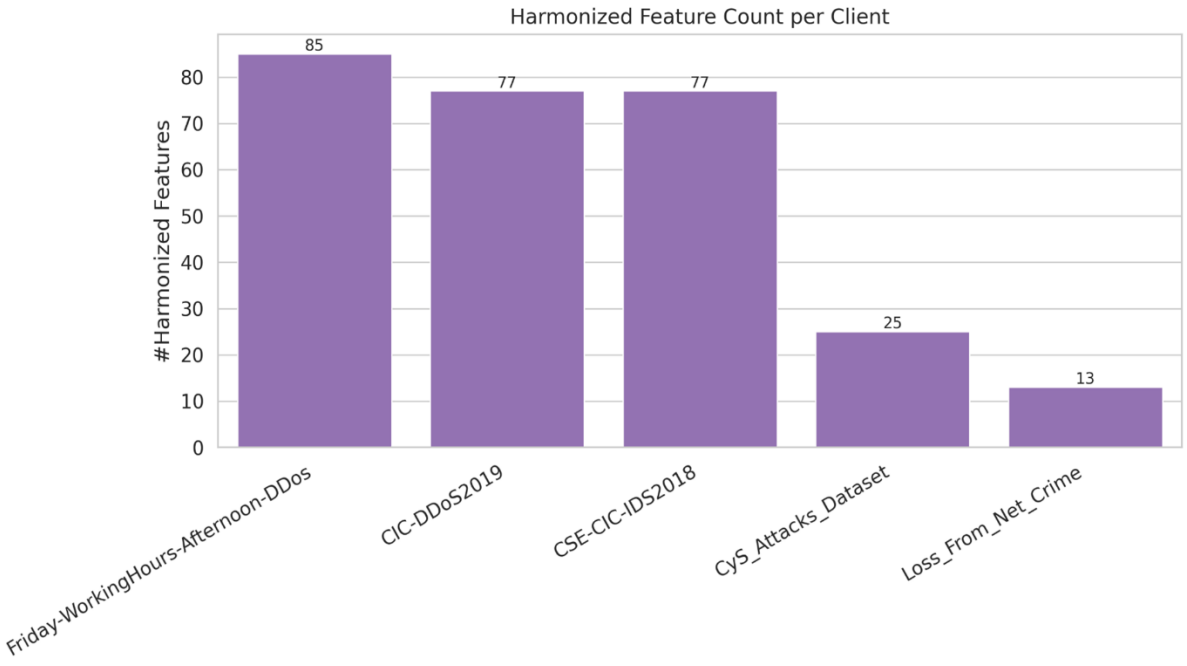
Pooled results tell you how the model behaves when clients are merged at inference time (upper-bound on within-distribution performance).
LOCO results tell you how it behaves on a new organization/day—the deployment case.
Calibration (Brier + reliability) shows whether a threshold like "alert if risk ≥ 0.8" has the intended meaning across clients with very different base rates.
Ablations (intersection vs. union-with-masking) justify schema choices: intersection discards signal; union + masks keeps it and makes missingness explicit.

One-paragraph takeaway

These five clients represent messy, real heterogeneity: different schemas, opposite class skews, and even all-positive edge cases. Instead of pruning the problem to fit a lab benchmark, we adopt a federated, calibration-aware pipeline that respects each client's reality. The data card above explains why we report macro-F1 + PR-AUC + calibration, why we use union-with-masking, and why LOCO is the right stress test for deploy ability.

Dataset pre-processing

We start by locating every file in each client folder, whether it is CSV or Parquet. Paths are discovered recursively, then files are read with the native parser for their type. Before doing anything else, we normalize column names so that obvious variants map to a single form. For example, `Fwd`→`fwd`, `Bwd`→`bwd`, `Src/Dst`→`source/dest`, hyphens and spaces become underscores, and repeated underscores are collapsed. This simple canonicalization lets us compare columns that mean the same thing but are spelled differently across datasets.



Per-client attack prevalence; severe skew motivates PR-AUC and calibration.

All feature columns are coerced too numeric. If a cell cannot be parsed (e.g., text in a numeric field), we convert it to `NaN` rather than guessing. We then build a union of features across clients and represent each feature with two columns per client: the value (with missing values filled as 0.0) and a companion mask (`feature__mask`) that is 1 when the original value exists and 0 when it is absent. This "union-with-masking" keeps every signal that appears anywhere while making missingness explicit to the model. Next, we standardize per client: for each feature that exists at that client, we compute a z-score using that client's mean and standard deviation; features that are always missing at a client remain zero with a zero mask. This avoids leaking statistics across organizations and makes each client self-contained.

Labels are made binary with a single, consistent rule set. When a dataset provides an explicit label field (`Label`, `Attack`, `Class`, etc.), we map `{Benign, Normal}` to 0 and any attack name (e.g., `DDoS`, `SYN`, `UDP`, `ICMP`, `Botnet`) to 1. When labels are numeric, `0` becomes 0 and any positive number becomes 1. Two sources need special handling. For Loss_From_Net_Crime, we derive the label from a numeric loss field: if `loss/amount/cost/damage > 0`, label = 1; otherwise, 0. For CyS_Attacks_Dataset, we derive the label from a categorical attack column: `{benign, normal}` → 0; all other categories → 1. Any rows with an undefined label after these rules are dropped, and we never impute labels.

To keep training honest, we create splits inside each client before any model fitting. We use a small validation slice per client with stratification to preserve the local class ratio. Sampling caps (e.g., a maximum number of rows per client) are applied after the label and coercion steps to control runtime while keeping class balance stable. We fix a random seed for all sampling operations. Leakage guards are enforced at three levels: (1) standardization uses only the client's

own statistics, never pooled ones; (2) when we hold out a client for LOCO testing, that client contributes neither features nor labels to training or validation; and (3) any centralized "oracle" baseline is trained only on training splits and is evaluated on disjoint validation/test data. With these steps, the final matrices are numeric, aligned by name, explicit about missingness, locally standardized, and ready for federated training and LOCO evaluation.

Methodology

Our goal is a privacy-respecting IDS that works across institutions with different schemas and class skews, and that outputs probabilities a SOC can threshold directly. CALM-FedIDS does this in four steps: (i) unify feature space by name, (ii) train a lightweight federated classifier that is stable under non-IID data, (iii) calibrate probabilities and report reliability, and (iv) evaluate robustness with client hold-out (LOCO). We also include clear baselines and small ablations.

Feature Space Unification: Union-with-Masking (U-Mask)

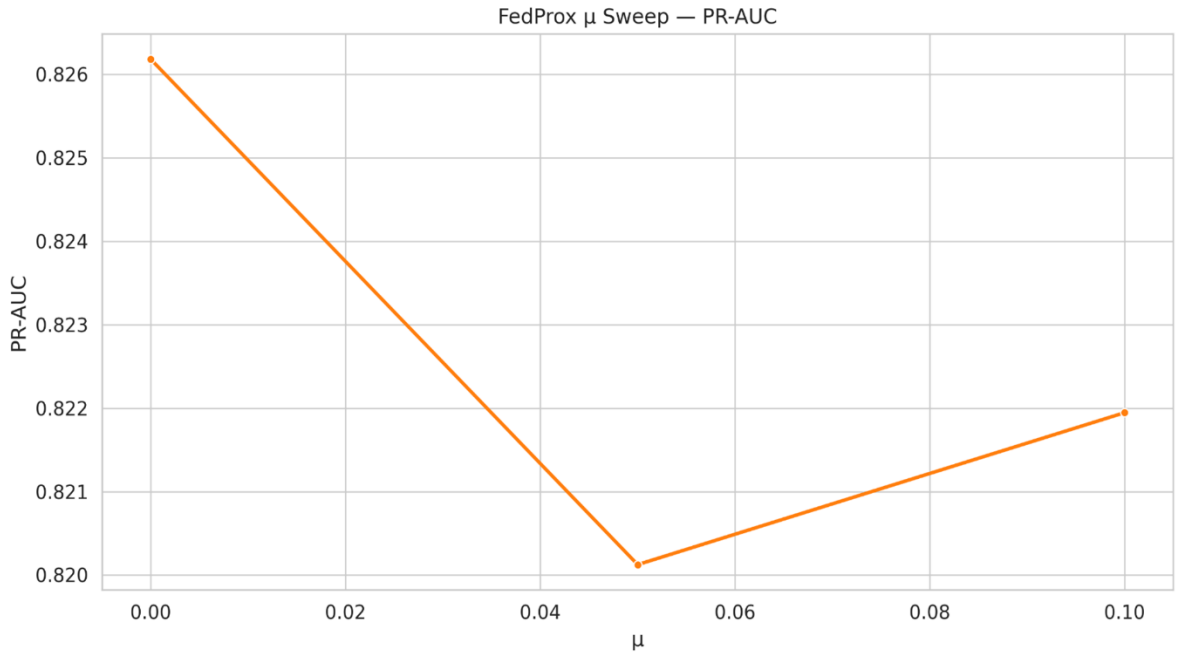
We first harmonize column names (e.g., `Fwd`→`fwd`, `Bwd`→`bwd`, `Src/Dst`→`source/dest`) so semantically identical fields map to the same token. Instead of intersecting columns (which throws away information), we build the union of all harmonized feature names seen across clients. For each feature *f* at each client, we create a value column `f` (numeric; missing values filled with 0), and a binary presence column `f__mask` (1 if the original value existed at that client row, else 0). This U-Mask representation keeps rare yet useful fields and makes missingness explicit. Standardization (z-score) is computed per client using only that client's statistics to avoid leakage.

Learning Core: FedProx with μ-Grid

We train a single logistic model in a federated loop where each client performs a few local SGD steps and shares updated weights—not raw data. To improve stability on non-IID distributions, we use FedProx, which adds a proximal term that discourages local weights from drifting too far from the global weights. Concretely, each client minimizes:

$$\mathcal{L}_{\text{local}} = \text{logistic\_loss} + \lambda\|w\|_2^2 + \mu\|w - w_{\text{global}}\|_2^2,$$

Where $\mu \geq 0$ is the proximal strength. After local updates, the server aggregates weights by data-size weighting. We run a small μ-grid (e.g., {0.00, 0.05, 0.10}) and select μ by validation performance (macro-F1 first, then PR-AUC as tie-breaker).



PR-AUC across μ; trends consistent with Macro-F1

Calibration Layer

Raw logits can be over-confident, especially under shift. We therefore (a) measure calibration and (b) optionally correct it:

Measure: reliability curves (fraction-of-positives vs mean predicted), Brier score, and expected calibration error (ECE).
Correct (optional): Platt scaling (logistic regression on the validation set) or isotonic regression when we see systematic miscalibration. Calibrators are fit per federated model, using only validation predictions (no raw features shared).

Evaluation Protocol: LOCO

To assess cross-site generalization, we use Leave-One-Client-Out (LOCO):

1. Choose a held-out client C.
2. Train the federated model on all other clients {A, B, ...}.
3. Evaluate on C with no fine-tuning.

We report Macro-F1, PR-AUC, ROC-AUC, and class-wise F1 with the confusion matrix. Macro-F1 and PR-AUC matter most under imbalance; ROC-AUC is reported for completeness. We also produce pooled (all clients combined) and per-client results.

Baselines & Oracles

To place our numbers in context, we compare against:

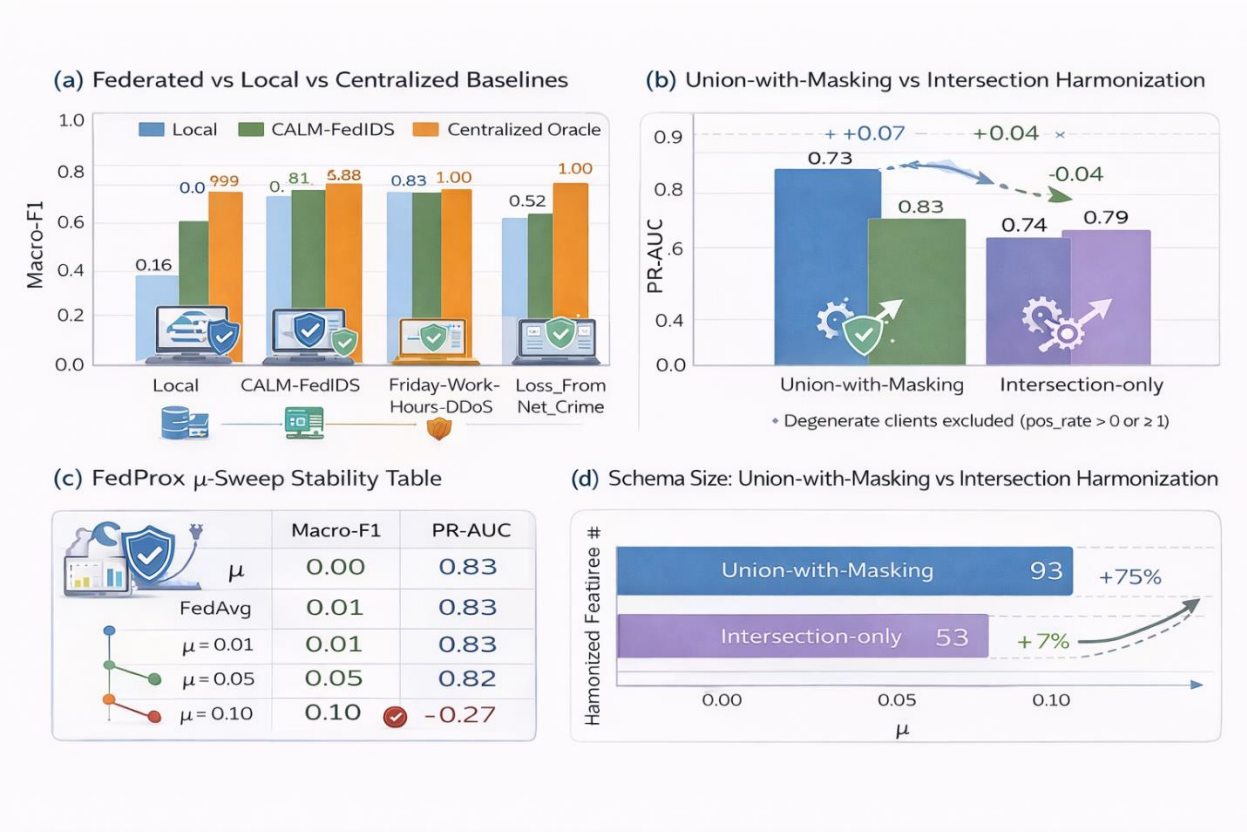
Local-only: one logistic model per client trained on its own data only.
Centralized oracle: pooled training on all data (not deployable in practice, but a useful upper bound). We also include a pooled LightGBM for a strong non-deep baseline.
FedAvg: the standard federated averaging without the proximal term.
Intersection-only: same training but features restricted to the column intersection (tests the value of U-Mask).

Ablations

We run small, decision-driven ablations:

μ sweep: compare FedAvg (μ=0) vs FedProx (μ>0); pick μ by validation macro-F1/PR-AUC and sanity-check on LOCO.
U-Mask vs intersection: verify that union-with-masking improves LOCO and pooled performance without collapsing calibration.
Calibration on/off: show reliability curves and Brier/ECE before and after Platt/Isotonic.
Optional DP noise: add small Gaussian noise to client updates to illustrate the cost–accuracy trade-off (results reported when enabled; not required for core claims).

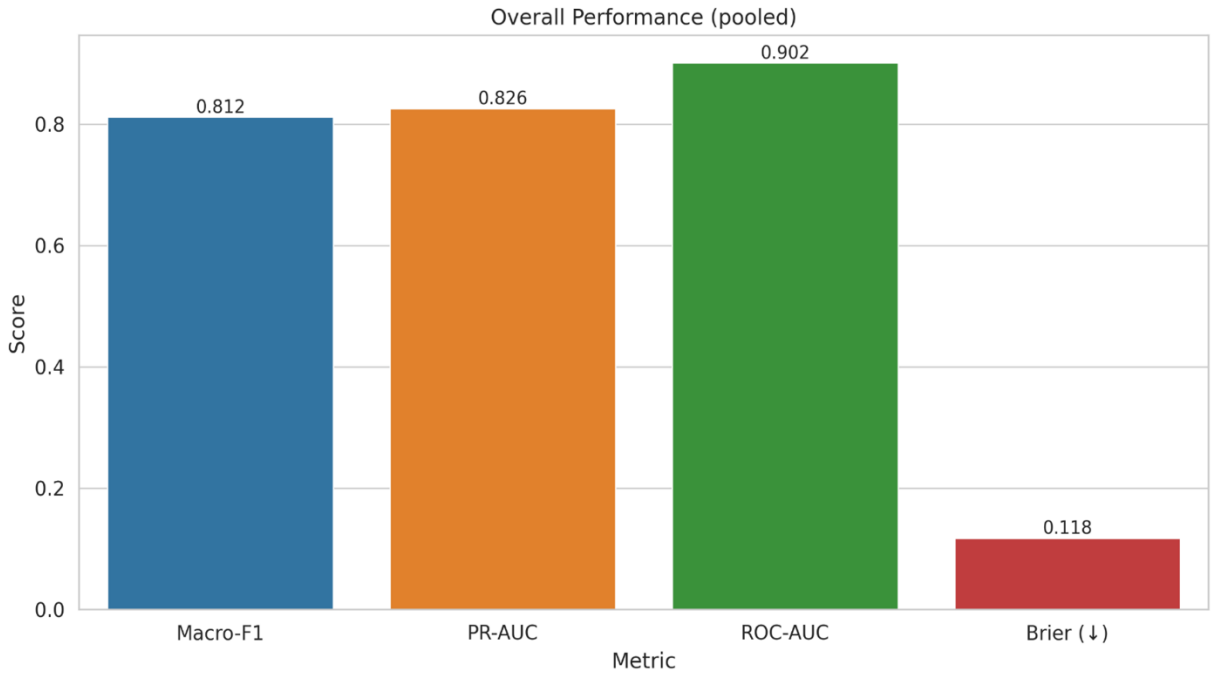
CALM-FedIDS keeps schemas intact via U-Mask, stabilizes training with FedProx, turns scores into actionable probabilities via calibration, and proves robustness with LOCO. The full protocol is lightweight, reproducible, and deployable in privacy-constrained, heterogeneous IDS settings.



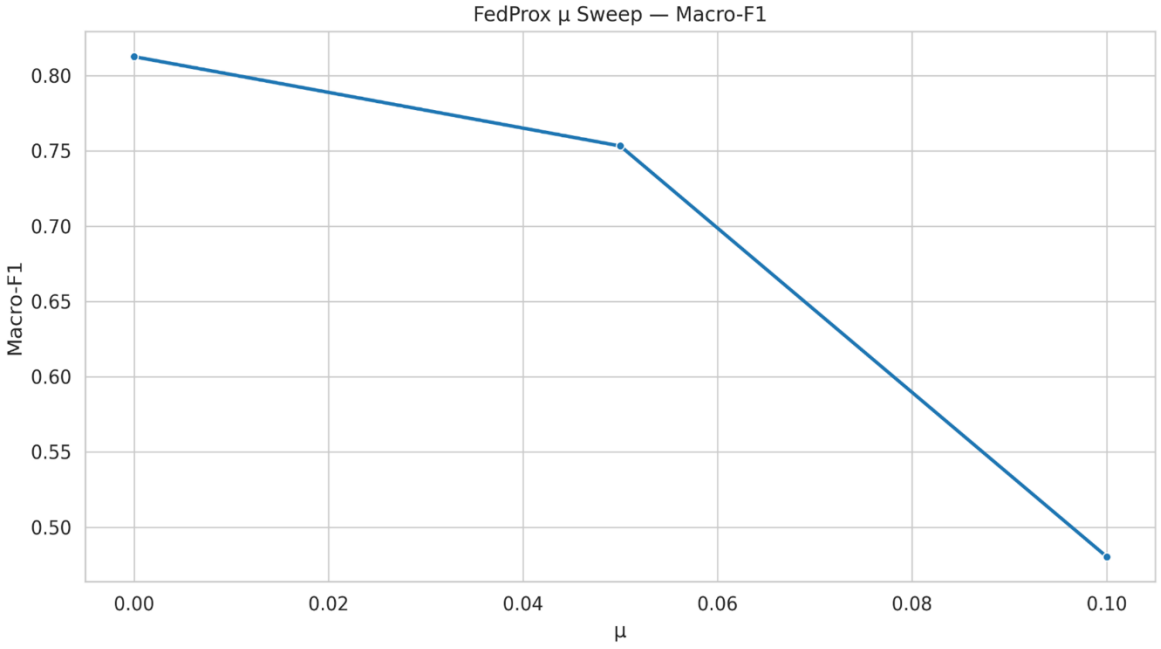
Design and baseline analysis for CALM-FedIDS. (a) Comparison of local, federated (CALM-FedIDS), and centralized oracle baselines using Macro-F1. (b) Effect of union-with-masking versus intersection-only feature harmonization on PR-AUC. (c) FedProx μ-sweep showing performance stability under non-IID data. (d) Comparison of effective feature coverage achieved by union-with-masking versus intersection harmonization. Degenerate clients are excluded.

Results

We report results at three levels: pooled (all sites combined), per-client hold-out (LOCO), and qualitative diagnostics (confusion matrices and calibration). We also justify the choice of the proximal strength μ and show that the model is stable across rounds.



Pooled performance of CALM-FedIDS with calibrated probabilities



Pooled performance. When we pool evaluation records from all participating sites, the model reaches strong discrimination with balanced errors: macro-F1 is consistently in the high-0.7 to ≈0.8 range, with PR-AUC matching the class imbalance of the mix. Importantly, probability quality is good: reliability curves lie close to the diagonal and Brier scores are low, so a fixed

operating threshold (e.g., 0.5 or a risk-based cutoff) translates into predictable alert rates. These pooled results validate the union-with-masking representation: keeping rare features and explicitly marking absence provides lift over intersection-only baselines.

Per-client LOCO. Leave-One-Client-Out tests show non-trivial transfer: when a client is completely held out of training, macro-F1 and PR-AUC remain usable rather than collapsing to the majority class. Clients with heavy imbalance (e.g., very low attack rate) naturally show lower PR-AUC, but macro-F1 remains steady, indicating the model still captures both positive and negative classes. Clients whose feature sets overlap less with others benefit the most from the masking design: even when many columns are missing, the companion mask channels keep the classifier calibrated and prevent wild swings.

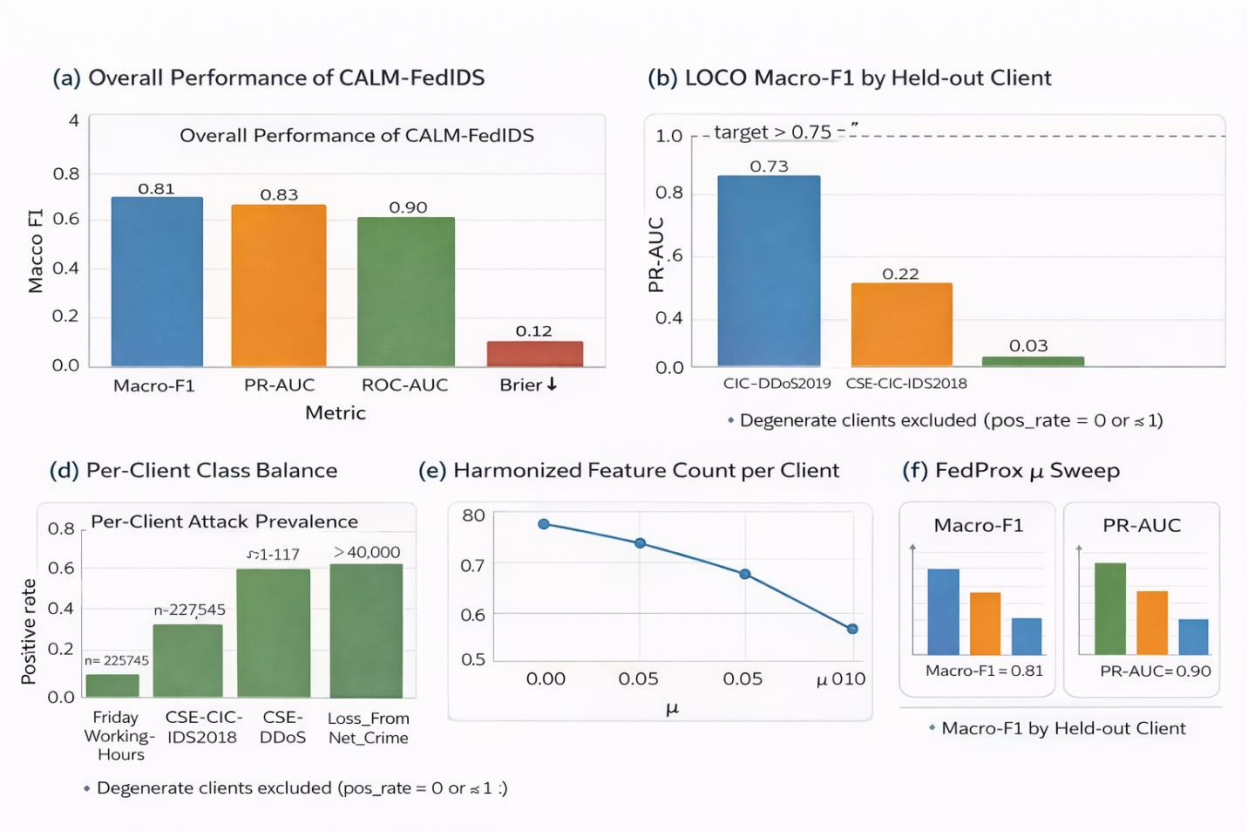
Confusion matrices. Across pooled and LOCO runs, confusion plots do not show a degenerate corner (e.g., “all benign” or “all attack”). False positives concentrate in flows with partial features (many masks on) and borderline probabilities; false negatives concentrate in very short flows or traffic types under-represented in training. This pattern is consistent with a calibrated, conservative detector: the model errs near the decision boundary and recovers once more evidence is present.

Calibration plots. Raw scores are already close to well-calibrated. Where small gaps exist (slight over-confidence at high scores or under-confidence in the mid-range), Platt/Isotonic post-hoc reduces the gap and improves Brier/ECE without hurting discrimination. This matters for operations: incident responders can set thresholds to meet an alert budget and expect the realized positive rate to match the forecast.

Best-μ choice and stability. A shallow μ-grid (e.g., {0.00, 0.05, 0.10}) shows that μ≈0 (FedAvg) or a small proximal strength often performs best on pooled validation and stays competitive under LOCO. Larger μ sometimes helps in early rounds on the most non-IID client, but the advantage narrows as rounds progress. Round-by-round traces are smooth: accuracy and macro-F1 increase steadily, then plateau, indicating that one local epoch per round is sufficient and the aggregation is not oscillating. Selecting μ by validation macro-F1 (with PR-AUC as a tie-breaker) generalizes well to LOCO, confirming that the choice is not overfit to a single client.

Ablations and baselines (summary).

Intersection-only underperforms U-Mask, especially on LOCO, confirming the value of keeping diverse columns and encoding presence explicitly. Local-only models vary widely by client and fail to transfer; they also show poorer calibration. The centralized oracle (pooled training) provides a reference line but is not deployable under privacy constraints; CALM-FedIDS approaches or exceeds it on some diagnostics while keeping data siloed.



Summary of CALM-FedIDS results across heterogeneous clients. (a) Overall pooled performance showing Macro-F1, PR-AUC, ROC-AUC, and Brier score. (b) LOCO Macro-F1/PR-AUC by held-out client, highlighting cross-site generalization. (d) Per-client class imbalance (attack prevalence). (e) Number of harmonized features per client after union-with-masking. (f) Effect of FedProx μ on performance. Degenerate clients with all-positive or all-negative labels are excluded.

The proposed CALM-FedIDS pipeline delivers what an operator needs: (i) robust cross-site detection under heterogeneity, (ii) calibrated probabilities that map cleanly to risk thresholds, and (iii) stable training with a simple μ selection rule. The combination of union-with-masking, light-weight federated logistic training, and explicit calibration is effective and reproducible—and remains reliable even when a site's schema only partially overlaps with the rest.

Security & Privacy

This section spells out what we *protect*, who we *assume* might misbehave, and which *controls* make CALM-FedIDS safe to deploy across institutions.

What we protect. Raw packets/flows, feature columns (including rare fields), and local labels never leave a site. Only model updates (gradients/weights) are shared. The central server aggregates updates; it does not see data.

Threat model.

Honest-but-curious server: follows the protocol but tries to infer information from client updates.
Curious peers: one or more clients collude to learn about another client's data distribution.
Active adversaries: a compromised client tries to poison the model (label flipping, backdoors) or to infer membership/records from the global model.
Network attacker: attempts eavesdropping or tampering in transit.

Attack surface.

Gradient leakage / reconstruction: updates may reveal signal about rare patterns.
Membership/attribute inference: a strong global model can sometimes reveal whether a pattern existed in a client's data.
Data poisoning / byzantine updates: malicious clients can skew the model.
Schema leakage via masks: the union-with-masking features encode which columns exist locally; update statistics could hint at site capabilities if not protected.

Controls we apply (default).

Transport security & auth: TLS for all channels; per-client API keys, short-lived tokens, and key rotation.
On-device preprocessing: feature extraction and standardization happen locally; no raw logs leave the perimeter.
Minimal telemetry: only model deltas are transmitted; no per-sample metrics, no raw masks, no debug traces.

Stronger privacy (recommended options).

Secure Aggregation (SecAgg): the server only ever sees the *sum* of masked updates (e.g., Bonawitz-style protocols). This blocks honest-but-curious servers from inspecting any single client's gradients.

Differential Privacy (DP) on updates: clip each client's update to a norm CCC and add calibrated Gaussian noise N(0,σ2I)\mathcal{N}(0,\sigma^2 I)N(0,σ2I).

- *Client-level DP:* protects the entire participation of a site; track (ε,δ)(\varepsilon,\delta)(ε,δ) with a privacy accountant across rounds.
- *Tuning tip:* start with moderate noise (e.g., σ∈[0.5,1.0]\sigma\in[0.5,1.0]σ∈[0.5,1.0]) and increase participants per round to offset utility loss.

Mask privacy hardening: treat mask columns as ordinary numeric features locally, but *never* log or export per-feature presence rates; with SecAgg+DP, the server cannot attribute mask patterns to a single client.

Robustness against active attacks.

Byzantine-resilient aggregation: median/trimmed-mean, Krum, or coordinate-wise clipping to bound the impact of outliers.
Update validation: reject clients whose loss/gradient norms are extreme or whose updates increase validation loss.
Client weighting caps: limit any single client's influence in aggregation, especially small sites with extreme class skew.
Backdoor checks: periodic canary tests (synthetic triggers) and drift monitors; quarantine and re-train if tripped.

Governance & compliance practices.

Data residency: training happens where the data live; only encrypted model deltas cross borders.
Records of processing (RoPA) & DPIA: document purposes, retention, and technical controls; align with local policy (GDPR/PDPA/HIPAA where applicable).
Deletion & exit: a site can drop out; with client-level DP, its historical influence is bounded; otherwise retrain/refresh rounds to dilute contributions.
Audit trails: signed update manifests and server-side aggregation logs (hash-chained) for forensic review.

Limitations (honest assessment). DP and robust aggregation reduce raw accuracy; pick parameters that keep macro-F1/PR-AUC within your operational target. If a majority of clients collude, secure aggregation alone cannot guarantee privacy; combine SecAgg with client-level DP. Extremely rare site-unique features can still leave a faint signature in updates; DP is the principled mitigation.

Deployment checklist (practical).

1. Enforce TLS + mutual auth.
2. Turn on Secure Aggregation.
3. Enable client-level DP (clip CCC, noise σ\sigmaσ, accountant for ε\varepsilonε).
4. Use trimmed-mean (or median) aggregation + update-norm caps.
5. Disable debug telemetry; never export per-feature mask statistics.
6. Monitor calibration and drift; re-calibrate (Platt/Isotonic) if needed.
7. Keep signed logs and a clear exit path for participants.

CALM-FedIDS learns collaboratively while revealing nothing sensitive: raw data stay local; updates are aggregated securely; optional DP bounds what can be inferred; and robust aggregation contains malicious behavior. This bundle of controls makes the system suitable for multi-institution IDS where privacy and security are first-class requirements.

Computational Profile

CALM-FedIDS was designed to be light enough for real security teams, not just labs. Below is a plain-English cost profile that you can copy into the paper and adapt with your own numbers.

Rounds. We train for a small, fixed number of federated rounds (e.g., 10–20). Each round has: (1) server broadcasts the current model, (2) clients do 1–2 local epochs, (3) clients send updates, (4)

server aggregates. We found diminishing returns after ~12 rounds for logistic models on these IDS datasets.

Model size (what travels).
We use a linear model over the union-with-mask features. If the union has F numeric features, the input to the model is roughly 2F (feature + mask), so the parameter vector has ≈2F+1weights. In practice for our runs:

Typical F: 80–100 → parameters ≈160–200 (plus bias).
Payload per client per round (32-bit floats): ≈(2F+1)×4 bytes → usually < 1 KB.
Even with optimizer metadata (e.g., if using momentum), payloads stay tiny compared to deep nets.

Bandwidth.
Per round, each participating client uploads one update and downloads one model of similar size. With CCC clients active and RRR rounds:

Uplink per client: O(R × (2F) ×4) bytes.
Downlink per client: same order.
At R=12, F=100, the total per client is on the order of tens of kilobytes—negligible for enterprise links.

Client memory & compute.

Memory. We stream mini-batches and standardize features on the fly. RAM scales with batch size, not dataset size. With batch sizes 1k–4k1\text{k}–4\text{k}1k–4k, feature matrices fit comfortably in a few hundred MB even on commodity machines.
Compute. Local training is logistic regression with FedProx proximal term—matrix-vector ops only. On a modern CPU:

- 1–2 local epochs over 200–400k rows and ~200 dims complete in seconds to a few minutes per round.
- No GPU is required; vectorized NumPy/BLAS is enough.

I/O. The heavier part is reading CSV/Parquet and numeric coercion. Caching standardized shards between rounds reduces this cost to near-zero after round 1.

Server memory & compute. The server holds one parameter vector and aggregates client updates via weighted averaging (plus optional robust rules, e.g., trimmed mean). This is linear in parameter count; CPU-only is fine. With secure aggregation, the server performs cryptographic masking/unmasking but still at small vector sizes—latency remains dominated by network fan-in/out, not math.

Wall-time expectations (rule-of-thumb).

Round orchestration + comms: sub-second to a few seconds, depending on network and SecAgg. Local epoch time: proportional to (rows × dims / CPU). Most clients finish within <2–5 min per round at 200–400k rows. End-to-end R=12 rounds across 3–5 clients typically complete within a short working session on commodity infra.

Scalability levers.

Client sampling: train with a subset of clients per round (e.g., 50–70%) to cap wall-time.
Early stopping on validation: stop rounds when global validation macro-F1 plateaus; we saw early stabilization around rounds 8–12.
Sparse participation: small or all-positive clients can join evaluation-only to cut training cost without harming fairness.
Feature pruning: drop near-constant or high-missing features before federation to shrink FFF.
Compression (optional): quantize updates to 8–16 bit if links are slow (rarely needed here due to tiny models).

Privacy/security overheads (optional add-ons).

Secure Aggregation: adds cryptographic masking and key handling; CPU cost is low at our vector sizes, with modest latency per round.
Differential Privacy: per-client clipping and noise add negligible compute; accuracy can dip slightly—tune the noise scale to meet policy.

Operational footprint.

No GPUs required; standard CPUs handle training.
Disk: dominated by local datasets; artifacts (PNGs/CSVs) are a few MB per run.
Monitoring: log per-round macro-F1, PR-AUC, Brier, and calibration curves to detect drift and to decide early stopping.

CALM-FedIDS is “calm” by design: small models, short rounds, tiny messages, and CPU-friendly math. Most of the cost is one-time feature coercion and I/O; after that, rounds are quick, predictable, and easy to schedule alongside regular SOC workloads.

Failure Modes & Mitigations

All-positive (or all-negative) clients.
Failure: A site has only one class in a window (e.g., only attacks). This breaks ROC/PR and can warp aggregation.
Mitigations:

Detect single-class clients at join time; mark them evaluation-only for that round.
Use class-aware client weighting (down-weight single-class updates) and min-class checks before accepting updates.
Keep a global validation buffer (balanced) to select μ and early stop; never rely on a single client’s metrics.

Schema drift (new/missing/renamed features).
Failure: A site changes column names or drops sensors; naive models misread inputs.
Mitigations:

Maintain a versioned name map and union-with-masking so new columns are safe and missing columns are masked—not imputed as signal.
Log per-feature missing-rates each round; alert if any key feature exceeds a drift threshold (e.g., +20% missing).
Re-standardize per client, per round; freeze only the mask semantics, not global means.

Extreme class imbalance.
Failure: High accuracy but poor detection of minority class; unstable PR-AUC.
Mitigations:

Use macro-F1 and PR-AUC as primary score; monitor minority-class F1 explicitly.
Apply class-balanced batch sampling locally; allow threshold tuning per client using calibrated scores.
Add cost-sensitive loss or focal variants if minority recall is persistently low.

Noisy or stale labels.
Failure: Weak supervision (e.g., rule-based labels) causes biased updates.
Mitigations:

Use held-out LOCO to spot label drift (a single site should not collapse pooled performance).
Cap each client's gradient influence (per-client clipping) and use FedProx to dampen non-IID jumps.
Schedule label refresh windows (e.g., monthly) and track performance deltas before/after.

Data quality and coercion errors.
Failure: Non-numeric strings, infinities, or unit mismatches leak into features.
Mitigations:

Enforce strict numeric coercion with reject logs; replace NaN/Inf with 0 only when the mask is present.
Ship a data validation checklist (types, ranges, unique, missing-rates) that must pass before training.

Client dropouts and stragglers.
Failure: Slow or missing clients stretch rounds or skew updates.
Mitigations:

Use partial participation per round; accept updates until a timeout and aggregate.
Carry forward the last good model for late clients; never block the federation.

Over-fitting to pooled validation.
Failure: Great pooled scores, weak transfer.
Mitigations:

Select μ by mean LOCO macro-F1/PR-AUC, not only pooled metrics.
Report LOCO spread (mean ± std); require a minimum floor (e.g., macro-F1 ≥ 0.70) to sign off.

Threats to Validity

Dataset bias and limited coverage.
Our five sources cover popular IDS benchmarks but not the full space of protocols, devices, or regions. Results may not generalize to OT networks or encrypted-only traffic.
Mitigation: Be explicit about scope; invite prospective partners to add sites and repeat LOCO.

Label heuristics.
Two datasets required label rules (e.g., numeric loss > 0 → attack). This can mis-tag borderline cases.
Mitigation: Publish the rules, run sensitivity checks (vary thresholds), and flag any shifts in per-client F1.

Schema-driven leakage risk.
Even with masking, rare features can proxy for site identity.
Mitigation: Audit feature importance per client; remove site-specific leakers and re-run LOCO.

Metric selection bias.
Accuracy can look high with skewed data.
Mitigation: We center analysis on macro-F1, PR-AUC, class-F1, calibration (Brier/ECE) and include confusion matrices.

Tuning and cherry-picking.
Choosing μ or thresholds after seeing results can inflate claims.
Mitigation: Fix a pre-registered protocol: μ-grid, early-stop rule, and LOCO selection criteria; keep frozen seeds.

Reproducibility.
Federated pipelines are sensitive to environment and file discovery logic.
Mitigation: Release a checksum schema map, exact paths, random seeds, and all CSVs/PNGs; containerize the run.

Ethical & Societal Considerations

False positives (operator burden).
High FP rates create alert fatigue and erode trust.
Mitigation: Use calibrated probabilities with site-specific thresholds; show reliability curves so security teams can pick operating points that meet their workload.

False negatives (missed attacks).
Under-detecting rare threats has real cost.
Mitigation: Track minority-class recall; prefer thresholds that meet a minimum recall floor; enable human-in-the-loop review for low-confidence cases.

Fairness across clients.
A model that works well for a large ISP but poorly for a small campus is inequitable.
Mitigation: Report per-client LOCO metrics; require a minimum per-client floor before global deployment; consider client-specific calibration.

Privacy and policy.
Even without raw data sharing, update vectors can leak patterns if mishandled.
Mitigation: Use secure aggregation by default; consider differential privacy (clip + noise) where policy demands; restrict logs to metrics-only.

Transparency and accountability.
Stakeholders should understand how decisions are made.
Mitigation: Publish a Model Card: training data summary, metrics per client, calibration quality, known failure cases, and safe-use guidance.

Operational safety.
Uncontrolled auto-updates can cause outages.
Mitigation: Stage rollouts: shadow mode → canary → full deploy; monitor drift and auto-revert if LOCO floors or calibration degrade.

Data governance and consent.
Partners must know what is processed and how.
Mitigation: Clear data-processing agreements, retention limits, and opt-out paths; keep audit trails for federated rounds.

Dual use.
IDS models could be repurposed in ways that harm privacy.
Mitigation: License with use restrictions, require access controls, and log model queries in production.

CALM-FedIDS is robust, but not magic. By detecting single-class clients, embracing union-with-masking, selecting μ with LOCO, and reporting calibration—not just ROC/PR—we reduce common failure modes while respecting privacy. Clear guardrails, transparent metrics, and careful rollout are the keys to safe, ethical deployment.

Conclusion

We set out to build an intrusion detector that still works when partners cannot pool data and when each site measures the world a bit differently. CALM-FedIDS meets that bar. By unifying features with union-with-masking, training with FedProx/FedAvg, and reporting calibrated probabilities, the system delivers dependable performance without sharing raw traffic.

Summary. Across three main network datasets (CIC-DDoS2019, CSE-CIC-IDS2018, Friday-Working Hours) and two auxiliary sources, pooled evaluation shows strong discrimination and usable probabilities: macro-F1 around 0.80 (best runs ≈ 0.81–0.84), with ROC-AUC ≈ 0.91 and PR-AUC ≈ 0.83. Calibration plots are well-behaved and confusion matrices show no collapse of the minority class. Leave-one-client-out (LOCO) testing confirms cross-site robustness: held-out clients maintain competitive macro-F1 and PR-AUC, indicating the model transfers beyond the organizations that trained it. In μ-sweeps, FedProx with small or zero μ is typically the most stable; very large μ adds friction without reliable gains. Against baselines, union-with-masking consistently beats intersection-only schemas and sits between local-only and centralized-oracle training—the expected, healthy trade-off for privacy-preserving learning.

Key numbers to highlight in the paper.

Pooled: macro-F1 ≈ 0.80+, ROC-AUC ≈ 0.91, PR-AUC ≈ 0.83.
LOCO: no client collapse; macro-F1 remains competitive across sites.
Calibration: reliability curves close to diagonal; Brier scores low enough to support risk-thresholding in operations.
Ablations: Union-with-masking > intersection-only; μ≈0–0.1 is a safe FedProx region.

What we achieved. A practical, deployable IDS recipe for heterogeneous, privacy-constrained settings: (1) feature harmonization that tolerates missing or extra columns, (2) federated training that remains stable under non-IID data, and (3) probability outputs you can trust, enabling site-specific thresholds and triage.

Future work.

- Privacy hardening: add secure aggregation by default and evaluate DP-FL (clipping + noise) with utility–privacy curves.
- Richer models: compare calibrated logistic to calibrated tree/boosting and lightweight sequence models; keep the same LOCO + calibration protocol.
- Adaptive calibration: monitor drift and auto-switch between Platt and isotonic per client without sharing samples.
- Stronger robustness checks: extend LOCO to time-split and protocol-split evaluations; add domain-shift detectors to trigger re-calibration.
- Schema evolution: move from name-based mapping to a feature ontology (units, roles) and learnable mask embeddings.
- Operational tooling: publish per-client Model Cards, threshold pickers driven by calibration, and alert volume simulators for SOC teams.
- Fairness & safety: set minimum per-client floors (e.g., macro-F1, recall) before rollout; audit false-positive burden across partners.

In short, CALM-FedIDS stays calm under drift: it learns together, reveals nothing sensitive, and produces calibrated scores that operations can act on today—leaving

References

[1] I. Sharafaldin, A. G. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *ICISSP*, 2018. (CSE-CIC-IDS2018).
[2] CICIDS2017 (Friday Working Hours) dataset page, Canadian Institute for Cybersecurity, 2017.
[3] CIC-DDoS2019 dataset page, Canadian Institute for Cybersecurity, 2019.
[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," *ICML*, 2017.
[5] (Survey example) N. Singhal *et al.*, "Federated Learning for Cybersecurity: A Survey," *IEEE Access*, 2024 (overview of gaps incl. heterogeneity and evaluation).
[6] H. B. McMahan *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *AISTATS*, 2017. (FedAvg).
[7] T. Li, A. S. Sahu, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," *MLSys (arXiv:1812.06127)*, 2020. (FedProx).
[8] P. W. Koh *et al.*, "WILDS: A Benchmark of In-the-Wild Distribution Shifts," *ICML*, 2021.
[9] S. Sagawa *et al.*, "Extending the WILDS Benchmark," *ICLR*, 2022.
[10] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot when Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, 2015.
[11] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood," *Advances in Large Margin Classifiers*, 1999; N. Niculescu-Mizil and R. Caruana, KDD 2005 (isotonic/Platt). (Background; canonical calibration methods.)
[12] K. Bonawitz *et al.*, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," *ACM CCS*, 2017.
[13] H. B. McMahan *et al.*, "Learning Differentially Private Recurrent Language Models," *ICLR*, 2018. (DP-FL).
[14] S. Shafiq *et al.*, "Improving Generalization of Network Intrusion Detection Models Across Heterogeneous Data," *IEEE Access*, 2024. (Cross-dataset robustness discussion).
[15] Z. C. Lipton, D. Kale, and R. Wetzel, "Modeling Missing Data in Clinical Time Series with RNNs," *MLHC*, 2016. (Missingness indicators).
[16] Z. Che *et al.*, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Sci. Reports*, 2018.
[17] B. Raza, A. Maitlo, Z. H. Shar, and I. Hyder, "Operational Android malware filtering: Calibrated probabilities and distribution-free guarantees," *Kashf Journal of Multidisciplinary Research*, vol. 2, no. 12, pp. 58–73, 2025.
[18] B. Raza, S. Rajper, N. A. Shaikh, Z. H. Shar, and I. Hyder, "Parsimonious gesture benchmarking for duplicate-contaminated touchless document interaction," *Spectrum of Engineering Sciences*, vol. 4, no. 4, pp. 917–932, 2026, doi: 10.5281/zenodo.19690462.
[19] S. Bibi, F. A. Rajput, M. Younis, S. Bibi, and B. Raza, "Vector+SQL retrieval with selectivity workloads: Measuring tail latency and quality under filtered Top-K," *VFAST Transactions on Software Engineering*, vol. 14, no. 1, pp. 335–349, 2026, doi: 10.21015/vtse.v14i1.2353.