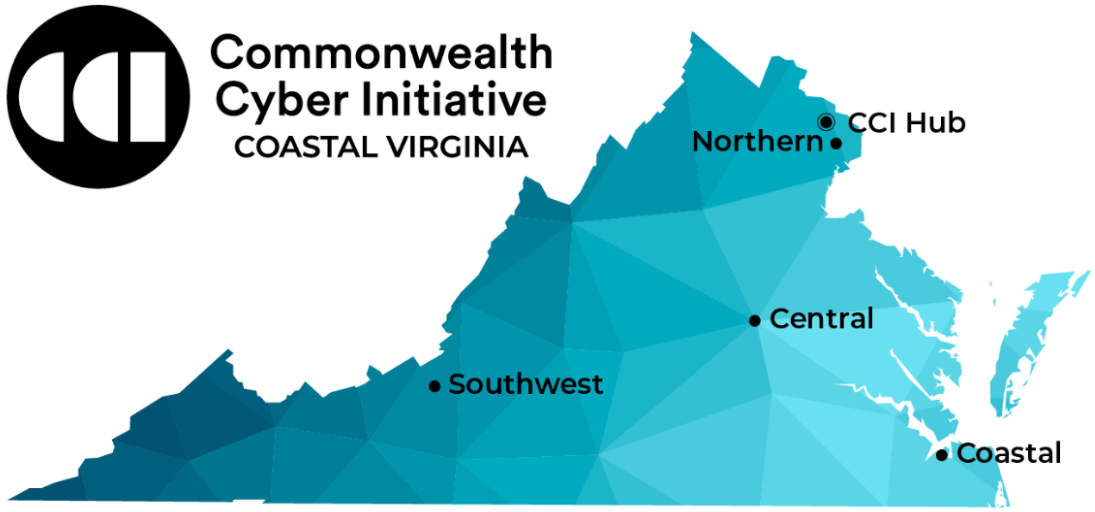
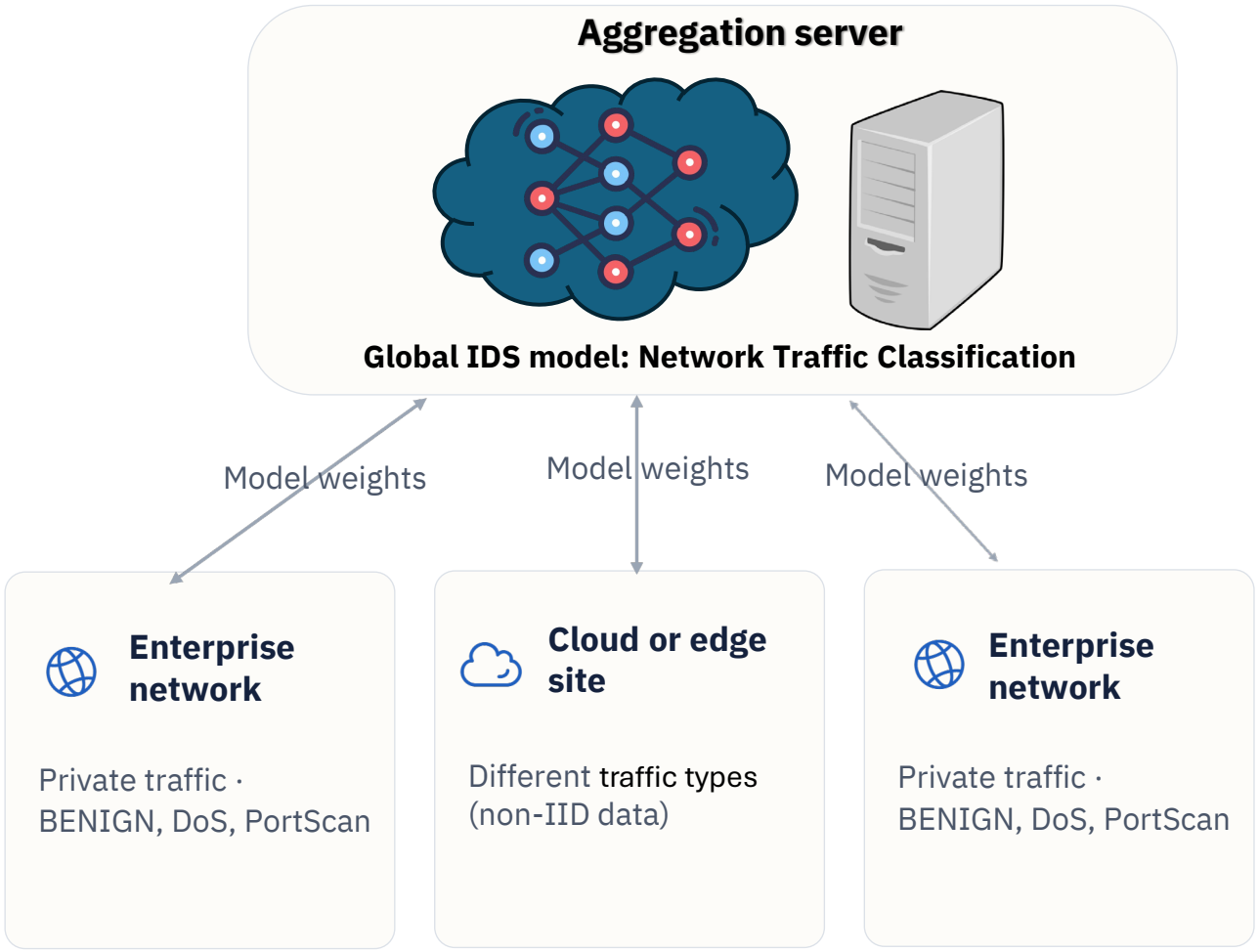
# CGAM-FL: Graph-Based Mitigation for Targeted Label-Flipping Attacks in Federated Intrusion Detection

Wael Khallouli[1], Rafael Diaz[1], Zeb Bowden[2], Biksah Chandra Singh[3], Yezhuo Li[1]

[1] School of Cybersecurity, Old Dominion University
[2] Virginia Tech Transportation Institute, Virginia Tech
[3] College of Science and Mathematics, California State University, Fresno, CA, USA



# Federated Learning for Intrusion Detection



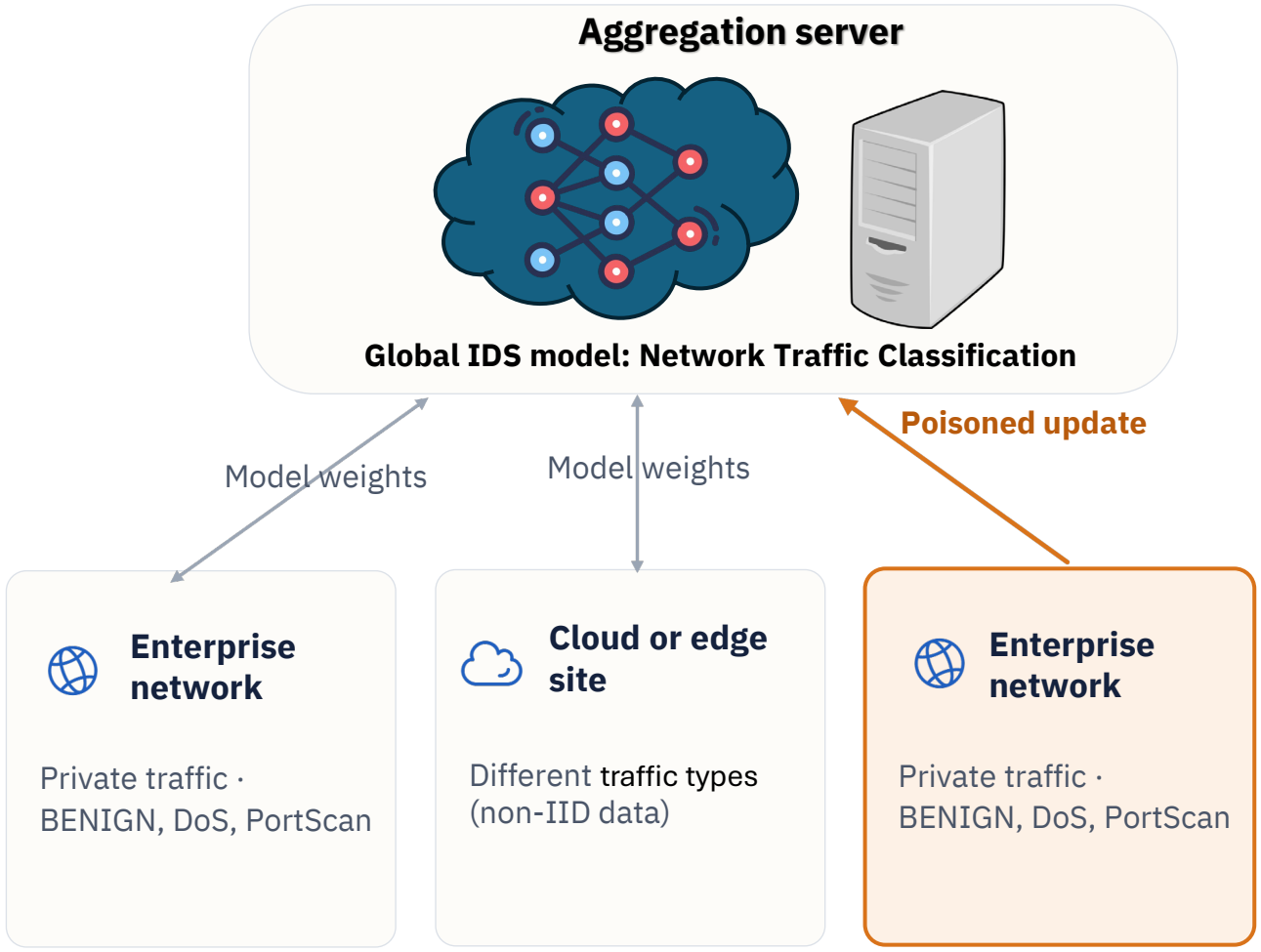

- Federated Learning allows multiple devices (clients) to collaborate on training a model without sharing raw training data
- Each participating client trains its own model locally and sends only model updates (gradients or weights) to a central aggregator.
- **FL-based IDS:** Organizations train a shared IDS model without sharing network traffic; only model weights reach the server.

## Why it matters

- ✓ Shared threat intelligence across organizations
- ✓ Learns from attack types that individual organizations rarely observe

# Research Problem

Mitigate Label-flipping attacks in FL-IDS to reduce detection evasion




**The risk:** one compromised client can teach the global model to miss an attack.

**Data poison attacks** (manipulation of training data):

- Label poisoning → attacker manipulates (flips) the labels in the training dataset
- Misleads the model into classifying source-class samples as an attacker-chosen target class

Label flipping hides attacks behind high overall accuracy

**Relabels DoS → BENIGN before training**

# Previous Work

X-FLSC: Two-step detection with parameter-level explanations

### Step 1 · Spectral clustering

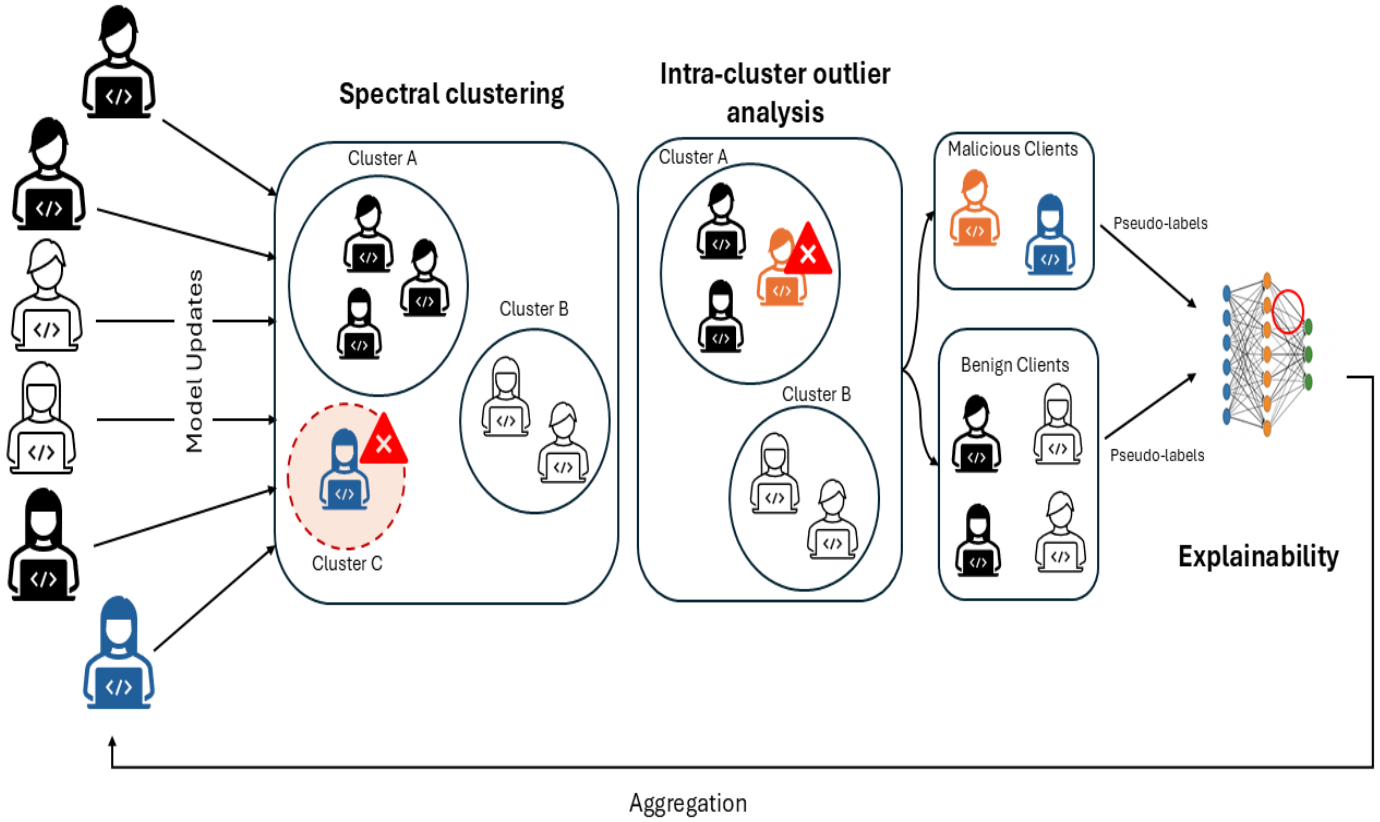
Mutual k-NN graph of normalized updates; the eigengap sets the number of clusters. Each cluster is scored by isolation and small size; clusters above a MAD-based threshold are flagged.

### Step 2 · Intra-cluster outliers

In the **remaining** clusters, flag clients far from their centroid (above median + 2.5 MAD), detect attackers hidden in benign-looking groups. FedAvg aggregates updates from the retained clients

### Post-hoc explainability

PCA (95% variance), then a surrogate decision tree trained on the detector's pseudo-labels; key components are mapped back to weights and ranked by AUC separability.




**Aggregator-side:** no validation data, no predefined number of clusters.

# CGAM-FL Key Idea

## WHAT PRIOR DEFENSES RELY ON

Most existing defenses, such as robust aggregation (Krum, Trimmed Mean) or clustering-based filtering (FLAME, LFighter), assume that benign clients form the majority or the largest group.

X-FLSC, like other clustering defenses, treats smaller, isolated groups as suspicious, so it assumes malicious clients are a minority. XFLSC can struggle in non-IID settings.

Most defenses inspect full, high-dimensional model updates.

## CGAM-FL's approach

**Label flipping** leaves a fingerprint in the classifier's class geometry.

**Relabeling DoS samples as BENIGN** can alter the similarity between their final-layer class-weight vectors.

CGAM-FL describes each client by the **cosine similarities between its class-weight vectors.**

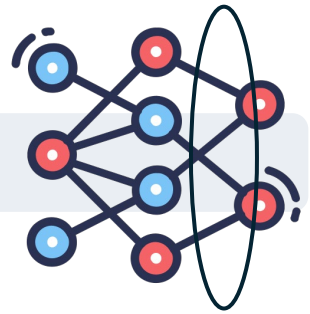
**Cross-client agreement:** a suspicious change counts only when several clients show it on the same pair of classes.

## Design constraints

- Server-side only, final layer only
- No server-side validation data
- No prior knowledge of the attacked classes
- No prior knowledge of the malicious fraction

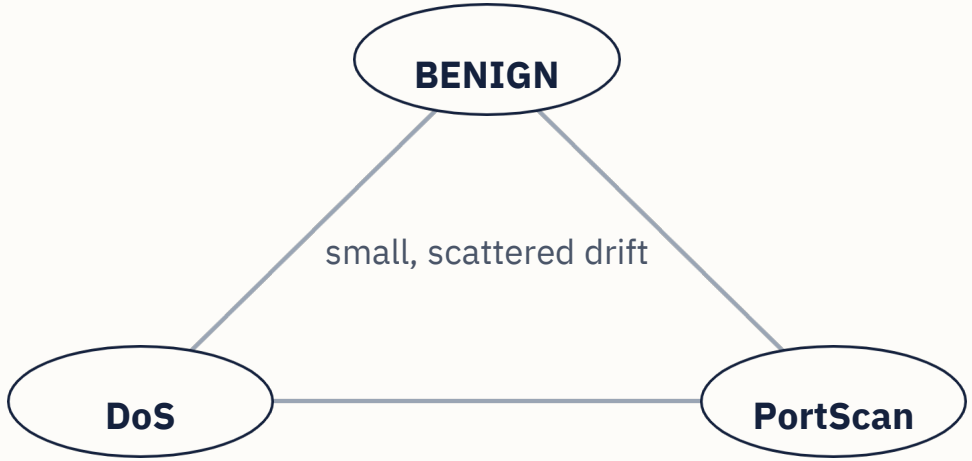
# Class-drift graphs

## Construct class-drift graphs

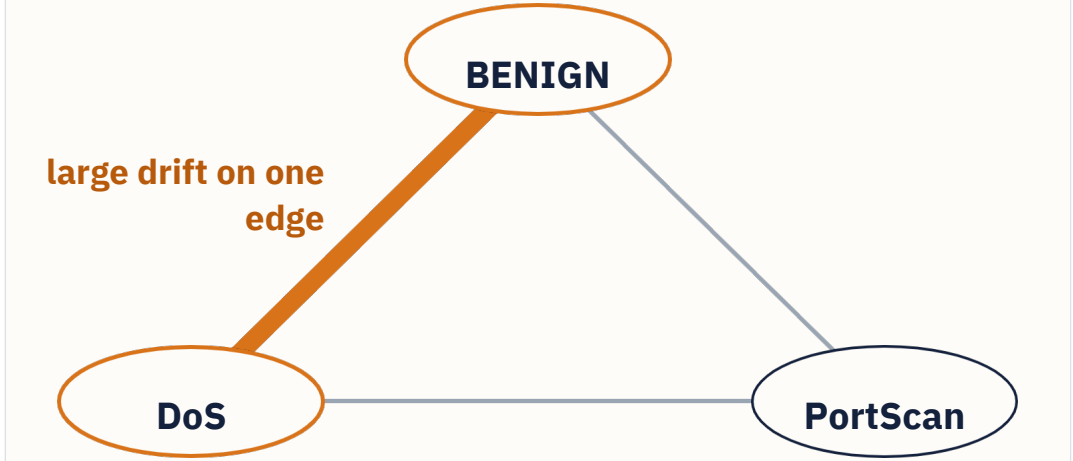


Drift $\delta(i, j) = \cos(\text{client class } i, \text{client class } j) - \cos(\text{reference class } i, \text{reference class } j)$

**Benign client**




**Poisoned client (DoS → BENIGN)**




The method requires a **clean reference model** collected before poisoning begins

# Proposed framework

## 1 · Self-referential scores

**Top edge and top node** per client, as **robust z-scores** within that client's own graph.

## 2 · Temporal smoothing and persistence count

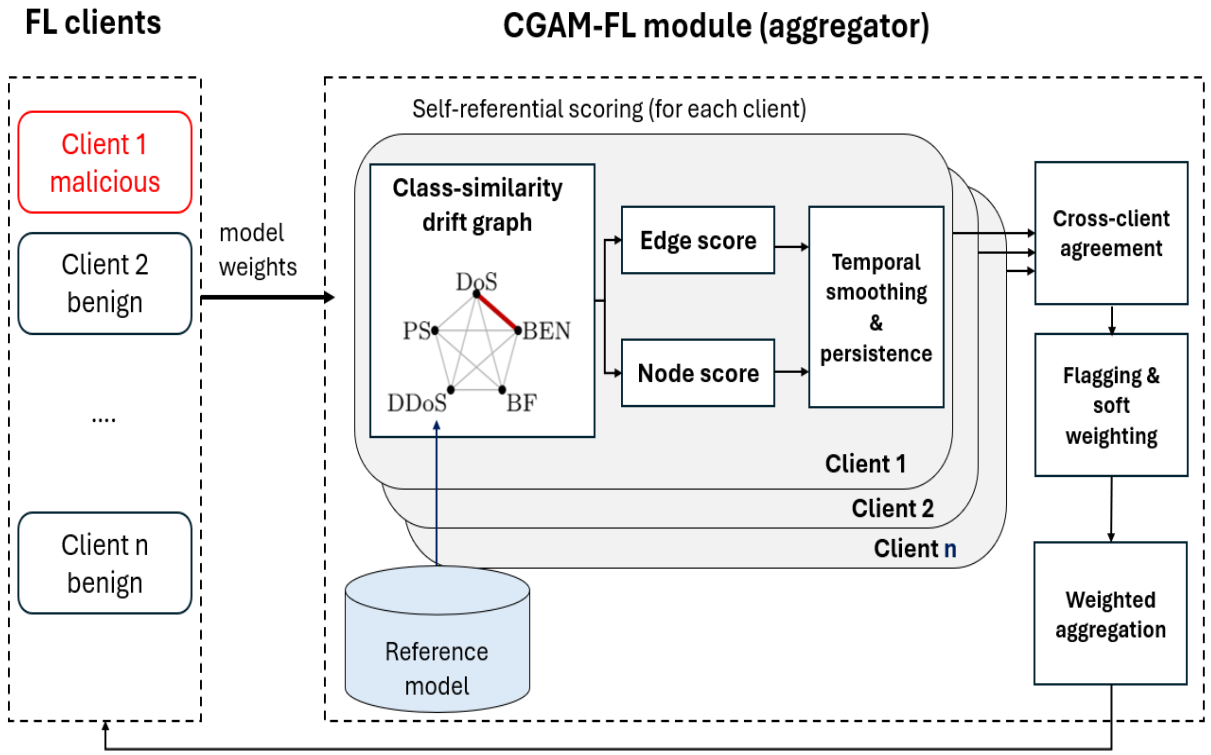
Scores are averaged over the last L rounds; **A client's persistence flag is set to true** when its **current top edge or node has appeared at least a minimum number of times** in the last P rounds.

## 3 · Cross-client agreement

A class pair (node) counts only when **enough self-flagged clients** ($z \geq 3.5$) **point to the same pair (node).** No honest-majority assumption.

## 4 · Flagging malicious clients

$$\begin{aligned}\text{flag}_k^t = &\left[\bar{z}_{\text{edge,t}}(k) \geq z_{\text{thr}} \ \wedge \ e_{k,t}^\star \in \mathcal{E}_t^{\text{corr}}\right] \\ \vee &\left[\bar{z}_{\text{node,t}}(k) \geq z_{\text{thr}} \ \wedge \ u_{k,t}^\star \in \mathcal{N}_t^{\text{corr}}\right] \\ \vee &\left[\text{persist}_{\text{edge}}^t(k) \ \wedge \ e_{k,t}^\star \in \mathcal{E}_t^{\text{corr}}\right] \\ \vee &\left[\text{persist}_{\text{node}}^t(k) \ \wedge \ u_{k,t}^\star \in \mathcal{N}_t^{\text{corr}}\right].\end{aligned}$$




## 5 · Soft aggregation

Flagged clients get weight s = max(0, 1 – (z – z_thr) / (z_max – z_thr)); unflagged clients keep weight 1.

# Experimental setup

**20**
clients, 30 rounds

**3**
data splits: IID, Dirichlet α = 0.3 and α = 0.1

**3**
poisoning rates: 20%, 40%, 60% malicious

**100%**
of source labels flipped

**Attacks:** single-label DoS → BENIGN; multi-label DoS + PortScan → BENIGN

**Baselines:** FedAvg (undefended), Krum, Trimmed Mean, LFighter, FLDetector

CICIDS2017 · feed-forward ANN · PyTorch + Flower · metrics averaged over the last 5 rounds

**Parameters:**

| Parameter | Symbol | Value |
|---|---|---|
| Reference-model round | $t_{\text{ref}}$ | $t_{poison} - 1$ |
| Drift-smoothing window | $L$ | 3 |
| Persistence window | $P$ | 5 |
| Minimum persistence count | $m_{\min}$ | 2 |
| Robust anomaly-score threshold | $z_{\text{thr}}$ | 3.5 |
| Cross-client agreement threshold | $\tau_a$ | 3 clients |
| Detection start round | $t_{\min} = \max(P, L)$ | 5 |

## CGAM-FL RESULTS

# CGAM-FL is most effective with 60% malicious clients and moderate moderate heterogeneity

Moderate non-IID (α = 0.3) · CICIDS2017 · last-5-round averages

| Method | Source acc. ↑ single | Source acc. ↑ multi | ASR ↓ single | ASR ↓ multi |
|---|---|---|---|---|
| FedAvg | 0.776 | 0.738 | 0.204 | 0.242 |
| Krum | 0.000 | 0.000 | 0.992 | 1.000 |
| Trimmed Mean | 0.000 | 0.000 | 0.852 | 0.915 |
| FLDetector | 0.713 | 0.530 | 0.266 | 0.440 |
| LFighter | 0.000 | 0.000 | 0.868 | 0.931 |
| CGAM-FL (ours) | 0.882 | 0.794 | 0.103 | 0.179 |

Best on all four metrics, and the highest global accuracy too (0.972 single, 0.991 multi).

CGAM-FL · Tables 2–3, 60% malicious

## CGAM-FL RESULTS

# IID Split at 60% poisoning rate: most models collapse. CGAM shows CGAM shows partial protection under this setting

**0.467**

CGAM-FL source-class accuracy, single-label (best baseline: Krum, 0.376)

**0.632**

CGAM-FL source-class accuracy, multi-label (every baseline: 0.003 or lower)

**0.351**

CGAM-FL ASR, multi-label (baselines: 0.698 or higher)

CGAM-FL · Tables 2–3, IID columns

# At 20–40% malicious and moderate moderate heterogeneity, gains are are small but consistent

α = 0.3 · each cell: source-class accuracy ↑ / ASR ↓

| Setting | CGAM-FL | FedAvg | FLDetector |
|---|---|---|---|
| 40%, single | 0.930 / 0.058 | 0.927 / 0.061 | 0.912 / 0.071 |
| 40%, multi | 0.959 / 0.031 | 0.926 / 0.064 | 0.915 / 0.070 |
| 20%, single | 0.975 / 0.021 | 0.950 / 0.041 | 0.949 / 0.041 |
| 20%, multi | 0.967 / 0.023 | 0.950 / 0.040 | 0.949 / 0.042 |

CGAM-FL · Tables 4–7, α = 0.3

# Where CGAM-FL falls short, and what comes next

## Limitations

**Extreme non-IID (α = 0.1):** at 60%, source-class accuracy 0.455 vs. 0.608 for FedAvg (single-label), because skewed benign clients drift too.

**Reference model** comes from an early round, a trade-off against full zero trust.

**Scope:** two fixed-rate label-flipping attacks, one dataset in the submitted version.

## Next steps

Adaptive, backdoor, model-replacement and Sybil attacks

UNSW-NB15 and other intrusion detection datasets for generalization

Temporal memory for slow, adaptive attackers

## Thank you.
## Questions?

Wael Khallouli · wkhallou@odu.edu