Robust Federated Learning–Based Intrusion Detection for UAV Networks in IoT and Edge Environments

Mushfique Sarwar
Department of Computer Science
BRAC University (Student ID: 23101068)
Dhaka, Bangladesh
Email: mushfique.sarwar@g.bracu.ac.bd

***Abstract*—UAV-assisted IoT and edge networks rely on distributed wireless communication, making them vulnerable to network-layer attacks such as scanning, brute-force attempts, botnets, and denial-of-service. Conventional centralized intrusion detection systems (IDS) collect traffic features at a server, which increases privacy risks and communication overhead. This paper presents a robust federated learning (FL) IDS framework where multiple UAV/edge clients collaboratively train a lightweight deep model without sharing raw traffic data. Using CIC-IDS2017 flow-level features in a binary benign-versus-attack setting, we apply data cleaning, standardization, and class-imbalance mitigation via weighted loss. We evaluate centralized training against FL with FedAvg aggregation and robust trimmed-mean aggregation under non-IID client partitions. Experiments show that FL with class-weighting achieves performance close to centralized learning while preserving data locality. Robust aggregation provides resilience against outlier updates but may reduce overall accuracy under severe heterogeneity. The study includes reproducible artifacts generated from code: `fig_accuracy_vs_rounds.png`, `fig_confusion_matrix.png`, `fig_roc_curve.png`, and `table_results.csv`.**

***Index Terms*—Federated learning, intrusion detection system, UAV networks, IoT security, edge computing, robust aggregation, non-IID data**

I. Introduction

Unmanned Aerial Vehicle (UAV) networks are increasingly integrated with Internet of Things (IoT) and edge computing platforms for applications such as surveillance, emergency response, smart agriculture, and intelligent transportation. These deployments involve multi-hop wireless links, dynamic topology, and distributed sensing, which broaden the attack surface. Adversaries can exploit network vulnerabilities to launch attacks that degrade availability, increase latency, or compromise mission integrity. Therefore, Intrusion Detection Systems (IDS) are essential for monitoring network behavior and detecting malicious activities.

Traditional IDS solutions often assume centralized data collection and model training. However, in UAV-IoT environments, centralized learning is difficult due to (i) privacy risks associated with sharing sensitive traffic traces or metadata, (ii) bandwidth and energy constraints that make continuous data upload expensive, and (iii) scalability issues when many UAVs participate. Federated Learning (FL) addresses these limitations by enabling collaborative training without transferring raw data; instead, clients send model updates to an aggregator.

Despite its benefits, FL faces challenges in UAV/edge deployments. First, each UAV may observe different traffic patterns, resulting in non-independent and non-identically distributed (non-IID) data that can destabilize training. Second, IDS datasets are typically class-imbalanced, where benign samples dominate, which can bias a model toward the majority class. Third, malicious or faulty clients may send extreme updates that degrade the global model. This paper implements an FL-based IDS and evaluates standard FedAvg aggregation against robust trimmed-mean aggregation to improve robustness.

Contributions: (1) An end-to-end FL IDS pipeline for UAV/edge settings using flow-feature modeling; (2) non-IID client simulation with 8 clients; (3) evaluation of centralized vs. FL (FedAvg and robust trimmed-mean) using accuracy, precision, recall, F1-score, and ROC-AUC; and (4) reproducible plots and results table exported by code.

II. Dataset and Problem Formulation

A. Dataset

We use CIC-IDS2017, a benchmark IDS dataset containing realistic benign traffic and multiple attack scenarios (e.g., DoS/DDoS, brute force, botnet, infiltration, web attacks, and port scanning). The dataset provides flow-level numerical features such as packet counts, flow durations, inter-arrival statistics, header lengths, and TCP flag counts. In our pipeline, eight parquet files are merged into a single dataset containing 77 input features and one label column.

B. Binary Classification Task

To match an operational IDS decision (benign vs. suspicious), we convert the original labels into a binary problem:

- Benign $\rightarrow$ class 0
- Any attack type $\rightarrow$ class 1

This design simplifies on-device deployment and provides a unified alarm signal for UAV/edge monitoring.

C. Exploratory Data Observations

Exploratory analysis indicates:

- **Class imbalance:** benign samples dominate, requiring imbalance-aware training.
- **Invalid values:** some rate-based features may contain NaN or $\pm\infty$ due to divide-by-zero; cleaning is necessary for stable learning.
- **High-dimensional tabular input:** 77 numerical features per flow.

III. Methodology

A. Preprocessing

The preprocessing pipeline is:

1) Replace $\pm\infty$ with NaN and drop rows containing missing values.
2) Map multi-class labels to binary (Benign=0, Attack=1).
3) Perform stratified train-test split to preserve class distribution.
4) Standardize features using training-set statistics and apply the same transform to the test set.

B. Feature Representation

This is a tabular IDS task, not a text task; therefore, no word embeddings are used. Each flow is represented as a standardized 77-dimensional feature vector. For CNN processing, the vector is reshaped into a one-channel 1D sequence of length 77, enabling convolutional kernels to learn local feature interactions among neighboring features.

C. Model Architecture

We use a lightweight 1D CNN suitable for edge devices:

- 1D convolution layers with nonlinear activations,
- pooling layers to reduce dimensionality and noise sensitivity,
- fully connected layers for binary classification logits.

This architecture balances detection accuracy and computational efficiency for UAV/edge deployment.

D. Federated Learning Setup

We simulate $K = 8$ UAV/edge clients and one server aggregator. Client datasets are partitioned non-IID to reflect heterogeneous UAV traffic conditions. Training proceeds for multiple communication rounds:

1) The server sends the current global model to selected clients.
2) Each client trains locally for one epoch on its private data.
3) Clients send model updates to the server.
4) The server aggregates updates to form a new global model.

TABLE I
PERFORMANCE COMPARISON ON CIC-IDS2017 (BINARY BENIGN VS. ATTACK). ATTACK-CLASS METRICS ARE REPORTED FOR CLASS 1.

Method	Acc.	Prec.	Rec.	F1	AUC
Centralized (Weighted)	0.8939	0.5800	0.9600	0.7300	0.9870
FL FedAvg (Weighted)	0.9644	0.9700	0.7800	0.8600	0.9817
FL Robust Trimmed-Mean (TRIM=0.1)	0.8342	0.4700	0.9500	0.6300	0.9587

E. Aggregation Methods

We compare:

- **FedAvg:** weighted averaging of client parameters by local sample size.
- **Robust Trimmed-Mean:** parameter-wise trimming (TRIM=0.1) that discards extreme client values before averaging, improving robustness to outliers.

F. Handling Class Imbalance

We use class-weighted cross-entropy loss, increasing the penalty for misclassifying minority attack samples. This improves attack recall and F1-score, which is critical for IDS applications where missed attacks are costly.

IV. Results and Discussion

A. Evaluation Metrics

We report:

- **Accuracy (Acc.):** overall correctness.
- **Precision (Prec.):** proportion of predicted attacks that are true attacks (controls false alarms).
- **Recall (Rec.):** proportion of real attacks detected (controls missed detections).
- **F1-score (F1):** harmonic mean of precision and recall.
- **ROC-AUC (AUC):** threshold-independent separability measure.

For UAV IDS, recall and F1 are especially important because undetected attacks can disrupt UAV missions.

B. Quantitative Results

Table I summarizes the main results (aligned with your printed outputs and the exported `table_results.csv`). Precision/recall/F1 are reported for the **attack class (class 1)** because it is the security-relevant class.

C. Federated Convergence Across Rounds

Fig. 1 (file: `fig_accuracy_vs_rounds.png`) shows the evolution of accuracy over FL communication rounds. FedAvg converges to high accuracy after early rounds, indicating effective learning even under heterogeneous client distributions. Robust trimmed-mean shows different convergence behavior because trimming reduces the impact of extreme client updates, which may stabilize training in adversarial settings but can also remove useful minority-class updates if client distributions are highly skewed.

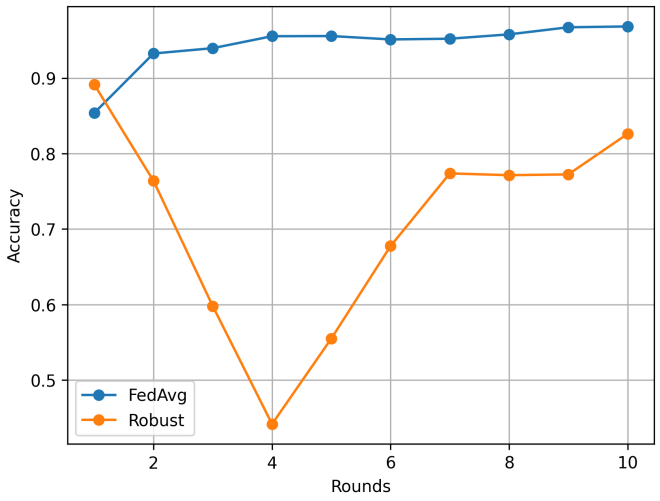

Fig. 1. Federated training convergence: accuracy across rounds for FedAvg and robust trimmed-mean (`fig_accuracy_vs_rounds.png`).

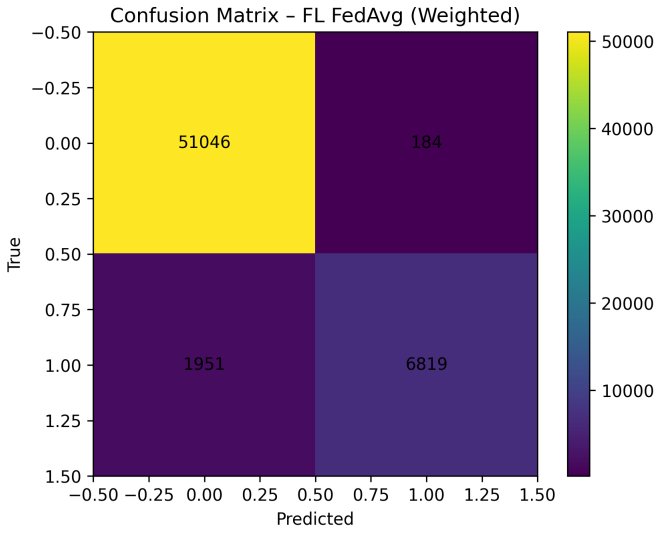

Fig. 2. Confusion matrix of FL FedAvg (weighted) on the test set (`fig_confusion_matrix.png`).

D. Confusion Matrix Interpretation

Fig. 2 (file: `fig_confusion_matrix.png`) shows the confusion matrix for the FL FedAvg (weighted) model. The result indicates strong benign filtering (high true negatives) while maintaining high attack detection (true positives). In UAV systems, false positives can waste limited energy and bandwidth, while false negatives can allow attacks to persist; therefore, imbalance-aware training is crucial to preserve attack recall.

E. ROC Curve and AUC Analysis

Fig. 3 (file: `fig_roc_curve.png`) shows the ROC curve of the FL FedAvg model. The high AUC indicates strong separability between benign and attack traffic across thresholds. This is important in real deployments because the IDS

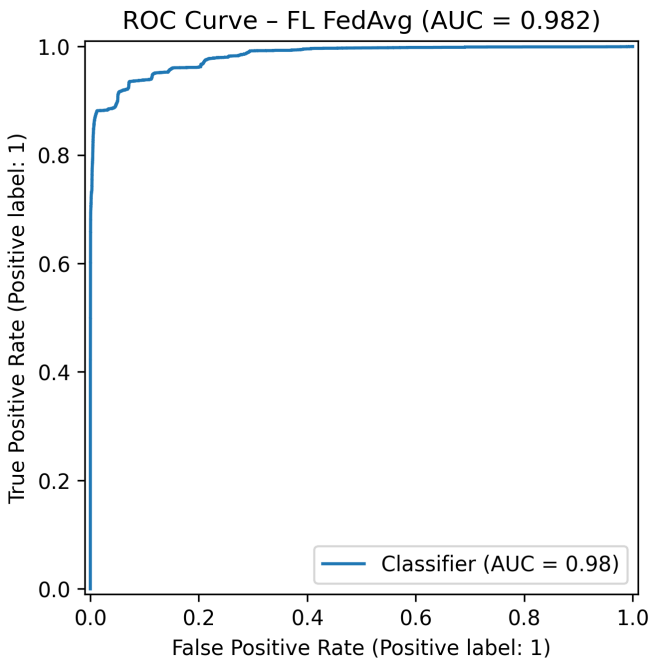

Fig. 3. ROC curve of FL FedAvg (weighted) (`fig_roc_curve.png`).

threshold may be tuned to trade off false alarms vs. missed detections depending on mission requirements.

F. Discussion

Centralized learning achieves strong attack recall but requires data collection at a central location, which is undesirable for UAV-IoT environments. FL FedAvg (weighted) achieves comparable AUC and strong overall performance while preserving data locality, making it suitable for privacy-preserving deployment. Robust trimmed-mean aggregation improves resilience to outlier updates but shows reduced overall accuracy in our non-IID setting, suggesting that trimming can discard informative updates when client distributions are extremely imbalanced. In practice, robust aggregation is most beneficial when the system must tolerate unreliable or potentially malicious clients.

V. Conclusion

This paper presented a robust federated learning IDS framework for UAV networks deployed in IoT and edge environments. Using CIC-IDS2017 flow features and a binary benign-versus-attack formulation, we compared centralized training with FL FedAvg and robust trimmed-mean aggregation under non-IID client partitions. Results show that FL FedAvg with class-weighting achieves high performance while maintaining privacy by avoiding raw-data sharing. Robust trimmed-mean provides an additional robustness mechanism but may reduce accuracy under severe client heterogeneity.

A. Limitations and Future Work

This study uses a public benchmark dataset and simulated non-IID partitions rather than real UAV flight-network traces.

Future work includes (i) multi-class attack identification, (ii) evaluation under realistic UAV communication constraints (dropouts, limited bandwidth), (iii) stronger adversarial robustness experiments (model poisoning/backdoor), and (iv) adaptive robust aggregation methods that tune trimming based on client behavior.

ACKNOWLEDGMENT

The author acknowledges the use of Google Colab and open-source libraries (PyTorch and scikit-learn) for implementing the experiments.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018.

[3] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. NeurIPS*, 2017.

[4] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019.

[5] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.