G^{2}FL: Robust Federated Learning for GNSS Spoofing Detection

Sheng Liu*, Wenjie Liu*, Ahmed Hussain*, and Panos Papadimitratos
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
Email: {shengliu, wenjieli, ahmhus, papadim}@kth.se

***Abstract*—Global Navigation Satellite Systems (GNSS) are vulnerable to spoofing attacks that can compromise critical infrastructure and safety-critical applications. While Federated Learning (FL) offers a promising paradigm for collaborative GNSS spoofing detection, with privacy preservation, existing proposals remain vulnerable to Data Poisoning Attacks (DPAs), by malicious clients manipulating local training labels to degrade global model performance. This paper presents G^{2}FL, a robust FL framework that detects and mitigates label-flipping attacks in GNSS spoofing detection. Our approach integrates two complementary mechanisms: a Gaussian Mixture Model (GMM)-based cluster analysis that distinguishes poisoned model updates from benign contributions, through validation prediction statistics, and a score-based client management system that maintains reliability assessments to exclude persistently malicious participants. Experimental evaluation using a real-world GNSS spoofing dataset demonstrates that G^{2}FL effectively mitigates DPAs, achieving 84.29% Area Under the Curve (AUC) compared to undefended FL (56.42%) and state-of-the-art defense, FoolsGold (65.80%), while maintaining performance close to attack-free scenarios (88.77%).**

***Index Terms*—Federated Learning, GNSS Spoofing Detection, Data Poisoning Attacks, Adversarial Defense, Cluster-based Detection, Privacy-Preserving Machine Learning**

I. Introduction

Global Navigation Satellite System (GNSS) provides essential positioning and timing services for critical infrastructures, including power grids, cellular networks, and connected and autonomous vehicle systems. However, the inherently low-power nature of GNSS signals renders them vulnerable to deliberate interference. While jamming attacks can deny GNSS services across geographical areas, spoofing attacks pose a more severe threat, as adversaries transmit falsified signals misleading receivers into computing incorrect position and time information.

The increase of GNSS attacks has motivated significant research in detection methodologies. Recent advances in Machine Learning (ML) and Deep Learning (DL) approaches have demonstrated substantial improvements over traditional signal processing techniques [1]–[3], leveraging large-scale datasets to identify sophisticated attack patterns. Simultaneously, participatory detection using consumer-grade devices has emerged as a scalable approach to distributed GNSS security monitoring [4], enabling collaborative threat detection across geographically dispersed participants.

Federated Learning (FL) [5] addresses the requirements of collaborative GNSS attack detection by enabling distributed model training across multiple devices while preserving data privacy, especially for security-critical applications [6], as it aggregates knowledge from numerous participants without centralizing sensitive location data. However, the distributed nature of FL introduces a severe vulnerability: *malicious participants can contribute poisoned data to compromise the global model and thus its spoofing detection capabilities*.

Data Poisoning Attacks (DPAs), notably label-flipping attacks, pose threats to FL-based systems. Malicious clients can manipulate their locally generated training labels to degrade model performance [7]–[10]. This manipulation is particularly accessible to attackers, as it requires only label falsification rather than sophisticated feature engineering. In GNSS spoofing detection, such attacks could systematically train the federated model to misclassify actual spoofing events as legitimate signals, fundamentally undermining system reliability. Existing FL frameworks for detecting GNSS applications have not adequately addressed DPAs, leading to:

> **Research Question:** *How can FL-based GNSS spoofing detection effectively identify and mitigate DPAs while preserving spoofing detection accuracy and maintaining robustness in non-Independent and Identically Distributed (IID) data environments?*

Answering this question requires addressing three key challenges: (i) distinguishing poisoned model updates from legitimate model diversity in heterogeneous client populations, (ii) managing client reliability over multiple training rounds without prematurely excluding benign participants, and (iii) maintaining spoofing detection performance under adversarial conditions. We address these aspects through G^{2}FL, a robust FL framework for GNSS spoofing detection, effectively mitigating DPAs. Our approach centers on two complementary mechanisms. First, we employ a Gaussian

*Equally contributing co-author.

Mixture Model (GMM)-based clustering to analyze the statistical characteristics of model updates, enabling the accurate distinction of poisoned contributions from benign model diversity in non-IID scenarios. Second, we utilize a score-based client management system that excludes persistently faulty and likely malicious clients, while gracefully handling occasional false positives from benign clients.

Contributions. (i) We formalize a threat model characterizing DPAs within FL-GNSS spoofing detection systems. **(ii)** We propose G^{2}FL, incorporating GMM clustering to accurately identify poisoned model updates through statistical analysis of validation predictions, **(iii)** We leverage a score-based client management system that adaptively maintains participant reliability scores for robust long-term defense, and **(iv)** Through experimental evaluation using real-world GNSS datasets, we demonstrate effective DPA mitigation while preserving spoofing detection accuracy, achieving 84.29% AUC compared to 56.42% under attack without defense.

Paper Organization. Section II reviews preliminaries and related work in FL, GNSS attack detection, and data poisoning defense mechanisms. Section III presents our system and adversarial model. Section IV details the G^{2}FL framework. Section V provides the experimental evaluation. Section VI concludes with future research directions.

II. Preliminaries and Related Work

A. Federated Learning

FL [5] enables collaborative model training while preserving data locality, and by extension privacy, through distributed parameter aggregation. Clients train models locally using their own private datasets and share only the model parameters with a central server, thereby eliminating the need for centralized raw data.

FL in Security Applications. The increase of Internet-of-Things (IoT) devices has motivated FL adoption for distributed security monitoring. Nguyen et al. [6] pioneered FL applications in IoT security through autonomous distributed systems leveraging device-specific communication profiles for anomaly detection. In privacy-sensitive medical domains, Sheller et al. [11] demonstrated that federated models trained across 10 institutions achieved 99% of centralized performance while maintaining strict privacy guarantees.

Security Challenges in FL. Despite inherent privacy advantages, FL systems remain vulnerable to adversarial attacks exploiting the distributed training process. Defenses have evolved to address these threats. CrowdGuard [12] introduced hidden layer analysis metrics that effectively identify poisoned models in non-IID scenarios by examining client data within secure enclaves. Metric-Cascades [13] employed multiple detection metrics, including Euclidean magnitude and directional analysis, to filter poisoned updates while maintaining minimal computational overhead.

B. GNSS Attack Detection

GNSS security has adopted ML and DL methodologies for signal interference detection. Contemporary approaches employ diverse frameworks including Support Vector Machine (SVM) [1], [14], Convolutional Neural Network (CNN) [2], [15], GMM [3], and Long Short-Term Memory (LSTM) [16].

Morales-Ferre et al. [1] achieved 94.9% jamming classification accuracy by framing the problem as image recognition through time-frequency analysis transformations. For spoofing detection, Borio et al. [2] utilized cross-ambiguity function analysis with CNN for satellite-specific detection, while Feng et al. [3] proposed unsupervised GMM-based approaches through intelligent clustering of position estimates. Kaasalainen et al. [17] introduced monitoring platforms integrating distributed infrastructure with DL for real-time signal quality analysis and service continuity assessment.

C. FL Data Poisoning Attacks and Defenses

FL systems face poisoning attacks where malicious clients manipulate local models [18] or falsify training data [7]. DPAs present particular challenges for resource-constrained platforms and prove harder to detect than model poisoning attacks, as servers cannot access client data for verification [8].

Existing Defense Approaches. Current countermeasures employ diverse strategies for poisoning detection. Distance-based methods like Krum [19] select updates with minimal summed Euclidean distances, while statistical approaches, such as TMean and Median [20], filter model parameters using robust aggregation. Similarity-based defenses include FoolsGold [21], which penalizes updates with high cosine similarity, and FLAME [22], combining differential privacy with clustering. FreqFed [23] applies the discrete cosine transform for frequency domain analysis of model updates.

Gap in Existing Solutions. There are limitations for FL-based GNSS attack detection. Existing methods [19], [21] analyze entire model parameters or output layers, but struggle to distinguish between poisoned models and benign ones in non-IID scenarios, where legitimate updates naturally exhibit high variance. Backdoor-focused defenses assume attackers falsify both features and labels, making them unsuitable for label-flipping attacks that only manipulate self-supervised labels. In the GNSS context, such attacks could systematically train the federated model to misclassify spoofing events as legitimate signals, thereby fundamentally undermining the detection system's reliability.

Furthermore, existing approaches (in the context of GNSS) lack integrated client-level management strategies that adaptively exclude malicious participants based on detection results. Recent FL-GNSS spoofing detection frameworks [24] demonstrated the feasibility of collaborative detection but require extensive training iterations to achieve convergence, a challenge that may be exacerbated under adversarial conditions. For FL-GNSS spoofing detection, the contradictory objectives between malicious and benign clients manifest in prediction distributions [8], presenting opportunities for statistical detection that current methods do not fully exploit.

III. System and Adversarial Model

This section discusses our system architecture and characterizes the threat model for FL-GNSS spoofing detection.

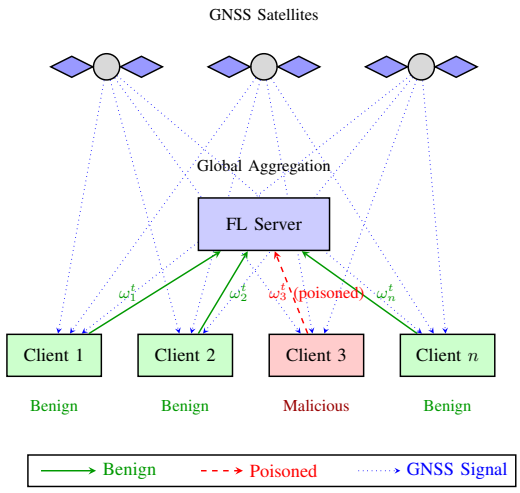

Fig. 1. DPAs in FL-GNSS. Malicious clients poison local data during training and submit falsified models to compromise global model performance.

A. System Model

Security and Privacy Protocols. Our FL framework ($\mathcal{F}$) operates within a distributed architecture comprising multiple participating clients (mobile devices denoted as $\mathcal{M}$) and a central coordinating server, as depicted in Fig. 1. The system accommodates a heterogeneous client population where a subset may exhibit malicious behavior while the majority maintain benign operation. Each client possesses pre-configured cryptographic credentials, including public-private key pairs and authentication certificates obtained from a trusted Public Key Infrastructure (PKI) and Certificate Authority (CA). This cryptographic foundation ensures secure communication channels and enables client authentication throughout the FL process.

FL Training Protocol. The operational workflow encompasses local training and global aggregation. During local training, each client performs model updates using its own privately held datasets, leveraging self-supervised learning to generate labels for GNSS spoofing detection. The model processes GNSS signal characteristics and positioning data to predict deviation measurements between geographical coordinates, enabling identification of spoofing attacks through anomaly detection.

In each communication round t, participant v_i updates the received global model ω^t to local model ω_i^t using its dataset D_i based on Eq. (1), where β denotes the learning rate and $\mathcal{L}_i$ is the loss function (e.g., Binary Cross-Entropy (BCE)).

$$\omega_i^t = \omega^t - \beta \nabla_{\omega^t} \mathcal{L}_i(\omega^t, D_i) \tag{1}$$

Following the completion of local training, clients transmit the computed model weights to the central server via secure communication channels. The server executes aggregation to form the updated global model ω^{t+1} according to Eq. (2), where q_i represents the aggregation weight, typically the normalized dataset size in FedAvg [25].

$$\omega^{t+1} = \sum_{S_i^t \in S^t} q_i \times \omega_i^t \tag{2}$$

This iterative process continues across multiple rounds until convergence, progressively refining the global model detection capabilities (while preserving individual client privacy). The converged model is subsequently deployed to participating clients for distributed GNSS spoofing detection.

B. Adversarial Model

The threat model considers internal adversaries, i.e., clients equipped with credentials and eligible to operate within the FL ecosystem, and contribute to GNSS attack detection. Adversaries cannot compromise the central server or other client devices. However, they can manipulate their local training processes and submit falsified model weights to the central server.

Malicious clients deliberately corrupt training labels to produce model weights that, although statistically plausible, contain biases designed to compromise the global model's integrity. For GNSS spoofing detection, adversaries flip labels from y_j to $1 - y_j$, systematically training the model to misclassify spoofing events as legitimate signals and vice versa. This label-flipping attack poses a particularly subtle threat, as poisoned contributions blend seamlessly with legitimate updates during the aggregation process.

The adversary seeks to undermine the effectiveness of spoofing detection by degrading the global model during inference. In that case, the degradation would be perceptible until an attack is mounted, harming any region or device in a system that relies on GNSS.

Rather than seeking immediate system compromise, adversaries pursue advanced persistent performance degradation, potentially targeting specific geographical regions or operational scenarios with stealthy DPA strategies and GNSS attacks. This undermines the reliability of Location-based Service (LBS), which relies on accurate GNSS data.

IV. Proposed Framework: G^{2}FL

This section presents G^{2}FL, our defense framework against DPAs targeting FL-based GNSS spoofing detection. The framework integrates three key components: (i) a self-supervised LSTM-based spoofing detection model, (ii) a GMM-based cluster analysis mechanism for identifying poisoned updates, and (iii) a score-based client management system for adaptive participant exclusion.

A. Base Model Architecture

Each client employs a self-labeling GNSS spoofing detector based on [24], utilizing LSTM [16] to process time-series data from GNSS receivers, network infrastructure, and onboard sensors. The model architecture, illustrated in Fig. 2, comprises two 100-unit LSTM layers followed by a fully

connected layer with Sigmoid activation, outputting spoofing probabilities in $[0, 1]$.

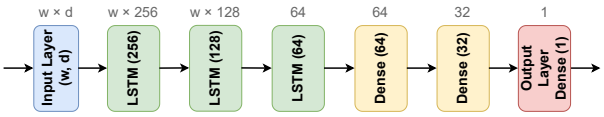

Fig. 2. LSTM model architecture for GNSS spoofing detection.

Feature Engineering. *Position-based features* (4 elements) capture residual and uncertainty of secure fused positions derived from GNSS, network, and Inertial Measurement Unit (IMU) data [26]. *Signal-based features* (32 elements) extract statistical properties (mean, median, minimum, maximum) from Automatic Gain Control (AGC), antenna Carrier to Noise Density Ratio (CN0), baseband CN0, and Doppler shift measurements for Global Positioning System (GPS) L1 and Galileo E1 signals. All features undergo preprocessing through extreme value removal (95th percentile), invalid entry imputation, and min-max scaling to $[0, 1]$.

Self-Labeling. Training labels quantify estimated GNSS position deviation as the normalized Euclidean norm of secure fused position residuals $\mu(m,t) - \mathbf{p}_{\text{gnss}}(m,t)$, preprocessed via capping and min-max scaling to $[0, 1]$ [26]. This self-supervised training approach avoids manually labeled attack data. The model optimizes BCE loss with adaptive learning rates and early stopping (20-epoch patience).

B. Cluster-based Detection

For each client i in round t, the server receives locally trained model parameters, w_i^t, and evaluates them on a validation set $(X_{\text{val}}, y_{\text{val}})$ to obtain predicted probabilities of data points $\hat{y}_i^t = f(w_i^t, X_{\text{val}})$.Since the server is trusted and cannot be compromised, the server-side validation prevents malicious clients from falsifying predictions. In real-world deployments, the server-side validation set can be sourced from verified secure GNSS measurements, e.g., with the help of infrastructure such as Continuously Operating Reference Stations (CORS).

Statistical Feature Extraction. We compute compact statistical features from validation predictions according to Eq. (3), as poisoned models typically exhibit altered probability distributions compared to benign models.

$$x_i^t = \left[\,\text{mean}(\hat{y}_i^t),\ \text{std}(\hat{y}_i^t),\ \text{median}(\hat{y}_i^t)\,\right]^\top \tag{3}$$

GMM Clustering. We fit a GMM with $K = 2$ components to the feature set $\{x_i^t\}_{i \in \mathcal{S}^t}$ through Eq. (4), partitioning clients into two clusters based on their prediction statistics.

$$z_i^t \sim \text{GMM}(2), \quad z_i^t \in \{0, 1\} \tag{4}$$

Each cluster $c \in \{0, 1\}$ receives an average validation AUC score calculated via Eq. (5).

$$\text{AUC}_c^t = \frac{1}{|\{i : z_i^t = c\}|} \sum_{i : z_i^t = c} \text{AUC}(w_i^t; X_{\text{val}}, y_{\text{val}}) \tag{5}$$

Algorithm 1 G^{2}FL: Cluster-based Misbehavior Detection with Score-based Client Management for FL

Initialize global model w^0 and scores $r_i^0 = r_{\text{init}}$
for $t = 1$ to T **do**
  The server broadcasts w^{t-1} to clients
  for each active client i **do**
    $w_i^t \leftarrow$ Local training based on Eq. (1)
    Compute predictions $\hat{y}_i^t$
    Extract features x_i^t based on Eq. (3)
    Compute validation AUC_i^t based on Eq. (5)
  end for
  Fit GMM($K = 2$) on $\{x_i^t\}$, assign cluster labels z_i^t
  Identify poisoned cluster $c^\star = \arg\min_c \text{AUC}_c^t$
  Flag set $P^t = \{i : z_i^t = c^\star\}$
  for each active client i **do**
    if $i \in P^t$ **then** $r_i^t \leftarrow \max(r_{\min}, r_i^{t-1} - \delta_{\text{penalty}})$
    else
      $r_i^t \leftarrow \min(r_{\max}, r_i^{t-1} + \delta_{\text{reward}})$
    end if
    if $r_i^t \le r_{\text{exclude}}$ and $|\text{Excluded}| < \frac{N}{2}$ **then**
      Permanently exclude client i
    end if
  end for
  Aggregation to obtain w^t based on Eq. (7)
end for

The cluster with lower AUC_c^t is identified as the *poisoned cluster*. Clients in this cluster are marked suspicious for round t and excluded from aggregation.

C. Score-based Client Management

To prevent occasional false positives from permanently excluding benign clients, we maintain client scores that evolve across training rounds, adopting the rating approach in [27]. Each client i has score $r_i^t \in [r_{\min}, r_{\max}]$, initialized as $r_i^0 = r_{\text{init}}$. Scores are updated according to Eq. (6) based on the detection results.

$$r_i^t = \begin{cases} \max(r_{\min}, r_i^{t-1} - \delta_{\text{penalty}}), & \text{if detected bad,} \\ \min(r_{\max}, r_i^{t-1} + \delta_{\text{reward}}), & \text{otherwise.} \end{cases} \tag{6}$$

When $r_i^t \le r_{\text{exclude}}$, client i is permanently excluded from subsequent training, subject to the constraint that fewer than half of all clients are excluded. After excluding detected and permanently banned clients, remaining models aggregate via FedAvg as shown in Eq. (7).

$$w^{t+1} = \frac{1}{|\mathcal{S}_{\text{agg}}^t|} \sum_{i \in \mathcal{S}_{\text{agg}}^t} w_i^t \tag{7}$$

Algorithm 1 summarizes the complete G^{2}FL procedure. This design achieves two objectives: *per-round misbehavior detection* identifies poisoned local models before aggregation, and *adaptive client management* excludes persistently malicious participants based on rating. Note that similar long-term management across several FL tasks can be achieved via revocation in PKI, but it is outside the scope of this paper.

V. Performance Evaluation

This section presents experimental evaluation of G^{2}FL using real-world GNSS spoofed data, demonstrating effectiveness against DPAs while maintaining detection accuracy.

Fig. 3. NSS data collection setup during Jammertest. *Left*: Recording infrastructure. *Right*: Android phones mounted for GNSS signal acquisition during drive-test traces.

A. Experimental Setup

Dataset. We utilize data included in the Jammertest 2024[1] NSS group field test campaign dataset; notably, 85 drive-test traces from six Android smartphones (Google Pixel 8, Pixel 4 XL, Xiaomi Redmi 9, and Samsung Galaxy S9) (Fig. 3) over a single day. Each device recorded timestamps, GNSS and network positions, IMU measurements, and GNSS signal descriptors (AGC, antenna CN0, baseband CN0, Doppler shift). The dataset used here was collected during GNSS spoofing/meaconing attacks. Ground-truth positions were obtained from two u-blox ZED-F9P receivers locked on benign constellations with assistance from a reference station. We allocate 10% of the traces for testing and 90% for training.

Implementation Details. The label-flipping attack is simulated during the federated learning: out of the six clients, two are designated as malicious, and they flip labels from y_j to $1-y_j$. The LSTM of the detector is implemented in Python 3.10 using Keras with TensorFlow backend. The architecture (Fig. 2) processes input sequences of dimension (w, d), comprising three LSTM layers ($256 \rightarrow 128 \rightarrow 64$ units) to progressively extract temporal patterns at multiple scales, two dense layers ($64 \rightarrow 32$ neurons with ReLU activation) implementing feature refinement through dimensionality reduction, and sigmoid output for binary classification probabilities.

We employ Adam optimization for its adaptive per-parameter learning rates suitable for non-IID federated data. The base learning rate $\eta_0 = 2 \times 10^{-3} \times \frac{bs}{36} \times \frac{N}{5.5 \times 10^4} \times \frac{1}{E_{\text{local}}}$ adapts to batch size $bs = 64$, training samples N, and local epochs E_{local} to maintain stable convergence across heterogeneous clients with varying data volumes. Exponential decay $\eta_t = \eta_0 \times 0.95^{\lfloor t/1000 \rfloor}$ gradually reduces learning rate to refine convergence. Gradient clipping (L_2 norm = 1.0) prevents instability from extreme gradients in adversarial scenarios, while early stopping (25-epoch patience, $\delta = 1 \times 10^{-5}$) balances convergence quality with computational efficiency. The model optimizes BCE loss $\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$, appropriate for probabilistic binary spoofing detection.

Training Configuration. We implement FedAvg aggregation on the central server with 150 total rounds and 1 local epoch per round. Performance evaluation employs Receiver-Operating Characteristic (ROC) curves and Area Under the Curve (AUC) metrics.

[1]https://jammertest.no/about/

TABLE I
COMPARATIVE PERFORMANCE OF DETECTION METHODS.

Method	AUC ↑
PDS [26]	83.49%
CL	85.32%
FL-Standard [25]	88.77%
FL-DPA	56.42%
FL-DPA-FoolsGold [21]	65.80%
G^{2}FL (Ours)	84.29%

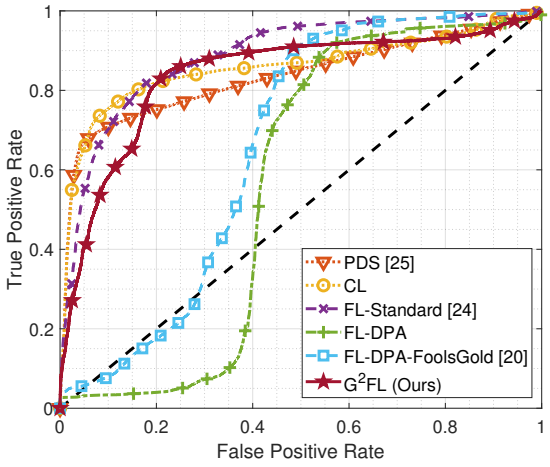

Fig. 4. ROC curves comparing various methods.

Method Comparisons. We evaluate six methods: **(1)** PDS [26], position-based detection; **(2)** CL, centralized learning; **(3)** FL-Standard [25], FedAvg without attacks; **(4)** FL-DPA, FedAvg under label-flipping attacks (Phones 2 and 6 as malicious clients); **(5)** FL-DPA-FoolsGold [21], FL-DPA with FoolsGold defense; **(6)** G^{2}FL, our proposed framework under label-flipping attacks.

B. Results Evaluation

Federated vs. Centralized Learning. Fig. 4 demonstrates that FL-Standard achieves superior ROC performance compared to both CL and PDS under benign conditions (without any DPAs). Table I quantifies this advantage: FL-Standard attains 88.77% AUC, outperforming CL by 2.45% and PDS by 5.28%. These results validate the effectiveness of federated collaborative learning for GNSS spoofing detection.

Vulnerability to DPAs. Poisoning attacks severely degrade detection performance. FL-DPA achieves only 56.42% AUC: a 32.35% reduction compared to FL-Standard (Fig. 4, Table I). Fig. 5 reveals that FL-DPA maintains consistently poor performance throughout training, confirming the severity of DPAs in FL-GNSS spoofing detection.

Defense Effectiveness. G^{2}FL demonstrates substantial improvement over existing defenses. While FoolsGold increases AUC from 56.42% to 65.80%, a 22.97% gap remains compared to FL-Standard. In contrast, G^{2}FL achieves 84.29%

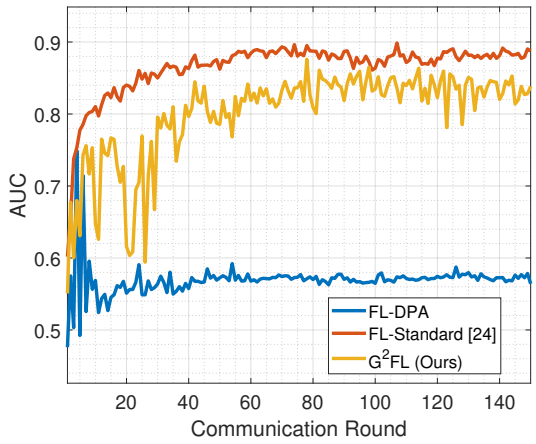

Fig. 5. AUC across communication rounds.

AUC, surpassing FL-DPA by 27.87%, exceeding FoolsGold by 18.49%, and approaching CL performance (85.32%). Fig. 5 shows that G^{2}FL exhibits initial fluctuations during the first quarter of training rounds as the system identifies and excludes malicious clients, then stabilizes to closely follow FL-Standard performance. The remaining 4.48% gap suggests opportunities for enhancement through global model correction techniques beyond local poisoned model filtering.

VI. Conclusion

This paper addresses the vulnerability of FL-GNSS spoofing detection to DPAs. We demonstrated that DPAs can severely compromise detection performance, reducing AUC from 88.77% to 56.42%, thereby undermining the reliability of collaborative GNSS security systems. To mitigate this threat, we proposed G^{2}FL, a framework that integrates GMM-based cluster analysis for detecting poisoned models and score-based client management for adaptive participant exclusion. Experimental evaluation with real-world GNSS spoofing data validates both the severity of DPAs and the effectiveness of G^{2}FL. Our framework achieves 84.29% AUC under attack conditions, representing a 27.87% improvement over undefended FL and an 18.49% improvement over state-of-the-art FoolsGold defense, while approaching attack-free performance levels.

Future work includes investigating (i) deployment considerations for the trusted server validation set, including its maintenance and size requirements, (ii) sensitivity to hyperparameters, (iii) robustness under adaptive attacks and varying proportions of malicious clients, and (iv) scalability through extended experimentation.

Acknowledgment

This work was supported in parts by WASP, VR, and in kind by the KAW Foundation, granting access to Alvis at the National Supercomputer Center.

References

[1] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, no. 22, p. 4841, 2019.
[2] Borhani-Darian et al., "Deep neural network approach to detect GNSS spoofing attacks," in *ION GNSS+*, Sept. 2020.
[3] Z. Feng, C. K. Seow, and Q. Cao, "GNSS anti-spoofing detection based on gaussian mixture model machine learning," in *IEEE ITSC*, Oct. 2022.
[4] Olsson et al., "Using mobile phones for participatory detection and localization of a GNSS jammer," in *IEEE/ION PLANS*, Apr. 2023.
[5] McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, Apr. 2017.
[6] Nguyen et al., "DÏoT: A federated self-learning anomaly detection system for IoT," in *IEEE ICDCS*, July 2019.
[7] Tolpegin et al., "Data poisoning attacks against federated learning systems," in *ESORICS*, pp. 480–501, Sep. 2020.
[8] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Lfighter: Defending against the label-flipping attack in federated learning," *Neural Networks*, vol. 170, pp. 111–126, 2024.
[9] S. Liu and P. Papadimitratos, "Safeguarding federated learning-based road condition classification," in *IEEE CNS*, pp. 1–9, Sep. 2025.
[10] S. Liu and P. Papadimitratos, "DEFEND: Poisoned model detection and malicious client exclusion mechanism for secure federated learning-based road condition classification," in *ACM SAC*, pp. 1–10, Nov. 2026.
[11] Sheller et al., "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, p. 12598, 2020.
[12] Rieger et al., "Close the gate: Detecting backdoored models in federated learning based on client-side deep layer output analysis," *arXiv preprint arXiv:2210.07714*, 2022.
[13] T. Krauß and A. Dmitrienko, "Avoid adversarial adaption in federated learning by multi-metric investigations," *arXiv preprint arXiv:2306.03600*, 2023.
[14] J. Xu, S. Ying, and H. Li, "GPS interference signal recognition based on machine learning," *Mob. Netw. Appl.*, vol. 25, no. 6, pp. 2336–2350, 2020.
[15] D. R. Kartchner, R. Palmer, and S. K. Jayaweera, "Satellite navigation anti-spoofing using deep learning on a receiver network," in *IEEE CCAAW*, June 2021.
[16] R. Calvo-Palomino, A. Bhattacharya, G. Bovet, and D. Giustiniano, "Short: LSTM-based GNSS spoofing detection using low-cost spectrum sensors," in *IEEE WoWMoM*, Aug. 2020.
[17] Kaasalainen et al., "Reason-resilience and security of geospatial data for critical infrastructures," in *ICL-GNSS*, June 2021.
[18] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, Feb. 2021.
[19] Blanchard et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NeurIPS*, p. 118–128, Dec. 2017.
[20] Yin et al., "Byzantine-robust distributed learning: Towards optimal statistical rates," in *ICML*, pp. 5650–5659, Jul. 2018.
[21] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *RAID*, pp. 301–316, Oct. 2020.
[22] Nguyen et al., "FLAME: Taming backdoors in federated learning," in *USENIX Security*, pp. 1415–1432, Aug. 2022.
[23] Fereidooni et al., "FreqFed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," in *NDSS*, Feb. 2024.
[24] W. Liu and P. Papadimitratos, "Self-supervised federated GNSS spoofing detection with opportunistic data," in *IEEE/ION PLANS*, Apr. 2025.
[25] McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, pp. 1273–1282, Apr. 2017.
[26] W. Liu and P. Papadimitratos, "Probabilistic detection of GNSS spoofing using opportunistic information," in *Proc. IEEE/ION PLANS*, Apr. 2023.
[27] P. Papadimitratos and Z. Haas, "Secure data communication in mobile ad hoc networks," *IEEE JSAC*, vol. 24, pp. 343–356, Feb. 2006.