Federated UE-Side False Base Station Detection via Location-RSRP Consistency

Hoonyong Park[1], I Wayan Adi Juliawan Pawana[2,3], Ilsun You[3,4]*

[1] Autocrypt Co., Ltd., Republic of Korea
[2] Department of Electrical Engineering, Udayana University, Indonesia
[3] Department of Cyber Security, Kookmin University, Republic of Korea
[4] KMU Global Research Center for ICT Convergence Security, Kookmin University, Republic of Korea
hoon4569@gmail.com, adijuliawanpawana@unud.ac.id, isyou@kookmin.ac.kr

Abstract

False base stations (FBSs) impersonate legitimate cells, making identifier-based detection unreliable. Representative attack data are scarce, while centralized collection of user equipment (UE) locations and reference signal received power (RSRP) measurements exposes sensitive information. This paper introduces location-RSRP consistency, the normal statistical relationship between a UE location and per-cell RSRP, and proposes a federated learning-based UE-side detector trained only on normal data. Regional clients retain raw measurements and collaboratively train a global regression model, which is distributed with region-specific thresholds for local detection. Network Simulator 3 experiments identified forward 1D-CNN-FedDyn as the preferred deployment configuration. Across three repeated runs, forward 1D-CNN-FedDyn achieved an F1-score of 0.987 ± 0.004, whereas forward Transformer-FedDyn achieved the highest F1-score of 0.993 ± 1.3×10^{-4}. Compared with Transformer-FedDyn, 1D-CNN-FedDyn used a 24.4 times smaller model artifact and provided 35.2 times higher inference throughput, offering the most favorable balance between detection performance and deployment efficiency.

Keywords: False Base Station Detection, Federated learning, Deep learning, Cellular network, Privacy

1 Introduction

Mobile networks constitute critical infrastructure, yet their radio access segment remains vulnerable to false base stations (FBSs), which impersonate legitimate cells to attract user equipment (UE) [1-3]. FBSs can collect identifiers, track users, force connection downgrades, inject messages, intercept traffic, or exploit baseband vulnerabilities [1, 4]. Recent vehicle-mounted phishing operations demonstrate that these threats remain practical [5]. Although 5G introduces identifier concealment and user-plane protection, legacy interworking, downgrade attacks, and specification-compliant malicious behavior continue to necessitate complementary FBS detection [1, 6-8].

Reliable detection requires evidence that remains valid when an attacker mimics legitimate identifiers and configuration parameters. UE locations and per-cell reference signal received power (RSRP) measurements exhibit a spatial relationship determined by base-station deployment and radio propagation, providing evidence beyond cell identifiers [3, 9]. However, representative FBS samples and reliable labels are difficult to obtain because attacks are rare and diverse. Centralized collection of location-RSRP measurements also concentrates sensitive mobility information [4, 10]. Moreover, geographic differences produce spatially non-independent and identically distributed (non-IID) UE data. A detection architecture must therefore learn from non-IID measurements without relying on attack labels or centralizing raw data.

Crowdsourced systems such as FBS-Radar and SeaGlass provide broad detection coverage but require centralized collection of location and radio measurements [4, 10]. Operator-side approaches, including Murat and 5G network-function-based detectors, analyze UE measurement reports and network information, but retain detection and processing on the network side [8, 11-13]. Location-based received-signal verification was introduced by Huang and Wang [9], while Nakarmi et al. evaluated normal-only anomaly detection using RSRP and location features [3]. These studies establish the usefulness of location-RSRP relationships but do not jointly support collaborative learning across spatially non-IID clients, local retention of raw data, and UE-side decisions.

To address this gap, this paper defines location-RSRP consistency as the normal statistical relationship between a UE location and its per-cell RSRP pattern. An FBS that impersonates a legitimate cell can increase the corresponding RSRP component without changing the UE location, thereby violating this relationship. Regional clients learn the normal relationship through federated learning (FL), and the server aggregates their model updates into a global regression model without collecting raw location-RSRP samples [14].

The proposed detector is trained exclusively on normal data. Each client calibrates a regional threshold, and the server distributes the resulting threshold map with the global model for local UE-side detection. Experiments

*Corresponding Author: Ilsun You; Email: isyou@kookmin.ac.kr
DOI: https://doi.org/10.70003/160792642026092705007

use datasets generated with Network Simulator 3 (NS3) [15] and compare MLP [16], 1D-CNN and 2D-CNN [17], GNN [18], Transformer [19], and XGBoost [20]. FedAvg [14], FedProx [21], SCAFFOLD [22], and FedDyn [23] are evaluated to analyze the effects of model architecture and federated optimization on detection performance and deployment efficiency. The contributions of this paper are as follows:

- A normal-only FBS detection system based on location-RSRP consistency, which is the statistical relationship between a UE location and per-cell RSRP under normal conditions, is proposed. The proposed system is designed to detect mismatches between location and RSRP patterns caused by an FBS impersonating a legitimate cell, without using FBS samples or attack labels for training.
- A federated learning architecture is designed in which spatially separated non-IID clients jointly train a normal regression model without transmitting raw location-RSRP samples to the central server. In addition, a threshold map derived from the normal calibration data of each region is distributed with the global model so that FBS decisions can be made on the UE side.
- The detection performance of MLP, 1D-CNN, 2D-CNN, GNN, Transformer, and XGBoost is compared using normal and FBS datasets generated with Network Simulator 3. In addition, convergence behavior and detection performance under FedAvg, FedProx, SCAFFOLD, and FedDyn are analyzed, and the computational and deployment efficiency of each model is evaluated.

The remainder of this paper is organized as follows. Section 2 reviews related work on FBS detection, and Section 3 presents the proposed system. Section 4 describes the datasets and experimental results and evaluates the performance and efficiency of the proposed system. Finally, Section 5 concludes the paper.

2 Background and Related Work

Section 2 is organized into three strands: (i) UE-side and signal-level FBS detection, (ii) network-based and crowdsourced detection, including machine learning based on radio measurements, and (iii) federated learning for wireless and cellular security.

2.1 UE-Side and Signal-Level Detection

UE-side FBS detectors include IMSI-Catcher-Catcher [24], EAGLE Security [25], Crocodile Hunter [26], and CellGuard [1]. These systems identify anomalies in cell identifiers, neighbor lists, connection downgrades, and signal strength; CellGuard additionally cross-checks cells against Apple's cell-location database. FBSDetector applies learning-based detection to multi-step attacks [27], while MODI converts Qualcomm diagnostic logs into structured NAS/RRC feature vectors for on-device analysis [28]. However, these approaches are limited to a single UE's observations, may require privileged access, and can produce false positives after legitimate network changes [7, 11].

Huang and Wang proposed a signal-level method that compares the measured synchronization-signal strength with the range expected from UE location, path loss, shadowing, and a base-station map [9]. Although this method uses location-radio consistency, each UE performs detection independently without collaborative learning.

2.2 Network-Based and Crowdsourced Detection

Crowdsourced systems extend detection coverage by combining observations from multiple devices. FBS-Radar geolocates suspicious cells using large-scale user measurements [4], while SeaGlass detects IMSI catchers through vehicle-mounted sensors [10]. However, both centralize sensitive location and radio data.

Network-based approaches instead analyze UE measurement reports (MRs). Murat compares the reported UE view with the network-expected view [11], while normal-only RSRP learning detected 75–95% of injected anomalies at a 0.5% false-positive rate [3]. Native 5G solutions include an in-CU specification- and XGBoost-based detector with detection rates of 99.6% and 96.75% [8], the finite-state-machine-based SMDFbs with 98% accuracy [12], and the supervised MRVF approach with 99% MLP accuracy [13].

Other studies address limited attack data through GAN-generated MRs [29] and quanvolutional learning for beyond-5G AI-RAN [5]. FBS localization using pairwise RSRP comparisons [2] and large-scale attack simulations [6, 30] have also been investigated. Although these studies demonstrate the value of radio measurements, many remain centralized or operator-centric and require aggregated MRs or labeled attacks [5, 8, 12-13]. Scalable UE-side detection without centralizing raw measurements therefore remains an open problem.

2.3 Federated Learning for Wireless Security

Federated learning (FL) enables a shared model to be trained from distributed device data by exchanging model updates rather than uploading raw samples [14]. For security-sensitive healthcare environments, Khan et al. proposed Fed-Inforce-Fusion, a federated reinforcement-based fusion model for intrusion detection and privacy protection in Internet of Medical Things (IoMT) networks [31]. This training paradigm is also applicable to UE-side FBS detection because observations from geographically distributed UEs can be incorporated while raw radio traces remain local. Crowdsourced coverage [4, 10] and MR-based features [3, 8, 13] can be combined with on-device preprocessing pipelines such as MODI [28]. However, geographic differences in UE locations and observed cells produce strongly non-IID client data, which can make stable global-model training difficult.

3 Proposed System

This section presents a federated learning-based UE-side system for detecting false base stations (FBSs) from

UE locations and per-cell reference signal received power (RSRP) measurements. Base-station deployment and radio propagation produce a characteristic RSRP pattern at each location. This statistical relationship between UE location and per-cell RSRP is referred to as location-RSRP consistency.

An FBS can violate this consistency by impersonating a legitimate cell and transmitting a strong signal. For example, when legitimate cells use equal transmit power, a UE in the nominal coverage area of cell A normally observes cell A as the strongest. If an FBS impersonates cell C, the UE may instead observe a cell-C-dominant RSRP vector while its location remains unchanged. This mismatch is defined as location-RSRP inconsistency and serves as evidence of an FBS.

The proposed system learns location-RSRP consistency using regression models. Each regional client retains its raw measurements and trains a local model, while the server aggregates model updates into a global model. After training, each client calibrates a regional threshold from normal data. The server distributes the global model, normalization metadata, and region-specific threshold map, enabling each UE to select the threshold for its current region and perform detection locally.

3.1 System Assumptions

A 5G radio access network (RAN) is considered in which each UE obtains its two-dimensional location (x, y) and a neighboring-cell RSRP vector arranged in a fixed cell order. Legitimate cells are assumed to use equal transmit power; therefore, normal RSRP differences primarily reflect UE–base-station geometry and radio propagation. An FBS is modeled as an unauthorized base station that impersonates a legitimate cell. A sufficiently strong FBS signal can violate the expected location-RSRP relationship and is treated as evidence of an FBS.

The service area is partitioned into K predefined regions, $\mathcal{R}_1$, ..., $\mathcal{R}_K$, with deterministic assignment at region boundaries. Client k represents the UEs and trusted normal data in region $\mathcal{R}_k$. Differences in location distributions and observed-cell patterns across regions result in non-independent and identically distributed (non-IID) client data.

Only regions included in federated training and provided with trusted normal calibration data are supported. All UEs in these regions, including newly deployed UEs, receive the same global model and region-specific threshold map. Other regions remain unsupported until their normal data are incorporated and their thresholds are calibrated. Poisoning [32] and Byzantine attacks [33] by malicious UEs are outside the scope of this study.

3.2 Input Representation and Location-RSRP Consistency

3.2.1 Location and RSRP Representation

Equation 1 defines the set of legitimate cells under observation.

$$\mathcal{B} = \{b_1, b_2, \ldots, b_M\} \tag{1}$$

In Equation 1, b_m denotes the mth legitimate cell, and M denotes the number of observed legitimate cells. The UE location associated with observation n is represented by the two-dimensional coordinate vector in Equation 2.

$$\mathbf{p}_n = [x_n, y_n]^T \in \mathbb{R}^2 \tag{2}$$

Equation 3 represents the corresponding per-cell RSRP measurements as a vector arranged according to the fixed cell order.

$$\mathbf{r}_n = \left[r_n^{(1)}, r_n^{(2)}, \ldots, r_n^{(M)} \right]^T \in \mathbb{R}^M \tag{3}$$

In Equation 3, $r_n^{(m)}$ denotes the RSRP measured by the UE for cell b_m. In this paper, the RSRP vector refers to the ordered array of measurements, whereas the RSRP pattern refers to the relative signal strengths and spatial structure contained in the vector. Equation 4 defines an indicator variable that distinguishes normal samples from FBS-affected samples.

$$\ell_n \in \{0,1\} \tag{4}$$

In Equation 4, $\ell_n = 0$ denotes a normal sample. In contrast, $\ell_n = 1$ denotes an FBS-affected sample in which the signal of the FBS is reflected in the measurement of the impersonated cell. This indicator identifies the normal samples used for regression training and threshold calibration; it does not serve as a target for binary classification. The proposed detector learns the regression relationship and its normal acceptance range using only normal samples.

3.2.2 Location-RSRP Consistency

Under normal conditions, the expected RSRP pattern at location $\mathbf{p}$ is represented by the location-conditional mean in Equation 5.

$$\mathbf{m}_R(\mathbf{p}) = \mathbb{E}[\mathbf{r} | \mathbf{p}, \ell = 0] \tag{5}$$

Equation 6 represents a normal RSRP vector as the sum of the location-conditional mean and a normal variation term. Equation 7 states that the variation is conditionally zero-mean.

$$\mathbf{r}_n = \mathbf{m}_R(\mathbf{p}_n) + \epsilon_n \tag{6}$$

$$\mathbb{E}[\epsilon_n | \ \mathbf{p}_n, \ell_n = 0] = \mathbf{0} \tag{7}$$

In Equations 6 and 7, ϵ_n represents the effects of propagation variability, measurement noise, NLOS propagation, shadowing, and other sources of normal variation. Conversely, Equation 8 defines the RSRP-

conditional mean location.

$$\mathbf{m}_P(\mathbf{r}) = \mathbb{E}\left[\mathbf{p} \middle| \mathbf{r}, \ell = 0\right] \tag{8}$$

Location-RSRP consistency holds when a pair $(\mathbf{p},\mathbf{r})$ agrees with the corresponding normal conditional relationship within an acceptable range of variation. This definition does not assume a deterministic one-to-one relationship between location and RSRP. RSRP measurements can vary at the same location, and similar RSRP patterns can be observed at different locations. As described in Section 3.3, the proposed system operationalizes location-RSRP consistency using detector-specific residual scores and thresholds calibrated from normal data.

3.2.3 FBS-Induced RSRP Perturbation

Suppose that an FBS impersonates legitimate cell b_j. Let $\mathbf{r}_n^N$ denote the normal RSRP vector that would be observed without the FBS, and let $r_{n,N}^{(j)}$ denote its jth component. The term $r_{\mathrm{FBS},n}^{(j)}$ denotes the received FBS signal strength measured under the same cell identifier. Equation 9 defines the nonnegative increase in the component of the impersonated cell.

$$\delta_n^{(j)} = \left[r_{\mathrm{FBS},n}^{(j)} - r_{n,N}^{(j)} \right]_+ \tag{9}$$

Equation 10 defines the positive-part operator, which sets a nonpositive argument to zero.

$$[a]_+ = \max(a, 0) \tag{10}$$

Equation 11 defines the resulting FBS-affected RSRP vector.

$$\tilde{\mathbf{r}}_n = \mathbf{r}_n^N + \delta_n^{(j)} \mathbf{e}_j \tag{11}$$

In Equation 11, $\mathbf{e}_j$ denotes a one-hot vector whose jth element is one. Equivalently, the jth component is given by Equation 12, which selects the stronger signal between the legitimate cell and the FBS.

$$\tilde{r}_n^{(j)} = \max\left(r_{n,N}^{(j)}, r_{\mathrm{FBS},n}^{(j)} \right) \tag{12}$$

In the following training, calibration, and detection formulations, $\mathbf{r}_n$ denotes the observed vector provided to the UE-side detector. Thus, $\mathbf{r}_n = \mathbf{r}_n^N$ under normal conditions, whereas $\mathbf{r}_n = \tilde{\mathbf{r}}_n$ when an FBS affects the measurement.

Because RSRP is measured in dBm on a logarithmic scale, Equations 9–12 do not represent physical signal superposition in the linear-power domain. Instead, these equations encode the threat-model assumption that the UE records the stronger signal when the legitimate cell and the FBS are observed under the same cell identifier. If the FBS signal is stronger, the FBS signal strength replaces the legitimate value in the component of the impersonated cell. Otherwise, $\delta_n^{(j)} = 0$, and the observed vector does not change. Because the actual UE location $\mathbf{p}_n$ remains unchanged during the attack, the deviation of $\tilde{\mathbf{r}}_n$ from the normal conditional relationship provides evidence of an FBS.

3.2.4 Client-Local Data Structure and Normalization

Let $\mathcal{K} = \{1,2,...,K\}$ denote the set of federated clients. Client k represents region $\mathcal{R}_k$, and every sample in its local dataset has a UE location within that region. Equation 13 divides the normal regional data of client k into training and calibration sets. Equation 14 states that these two sets are disjoint.

$$\mathcal{D}_k^{N,\mathrm{all}} = \mathcal{D}_{\mathrm{tr},k}^N \dot{\cup} \mathcal{D}_{\mathrm{cal},k}^N \tag{13}$$

$$\mathcal{D}_{\mathrm{tr},k}^N \cap \mathcal{D}_{\mathrm{cal},k}^N = \varnothing \tag{14}$$

Both sets contain only normal samples with $\ell=0$. The training set is used to optimize the regression model, whereas the calibration set is used to determine the threshold associated with region $\mathcal{R}_k$. The regional threshold is not a personalized parameter for an individual UE; every UE operating in the same region uses the same threshold entry.

The inputs and targets of the neural regression models are normalized to account for the different numerical scales of the location coordinates and RSRP measurements. Let $\mathbf{c}_p$ and $\mathbf{h}_p$ denote the center and half-width vectors of the target service area, respectively. The server obtains the global RSRP normalization parameters by aggregating region-specific sufficient statistics instead of raw samples. Client k reports the number of normal training samples $N_k = |\mathcal{D}_{\mathrm{tr},k}^N|$. The cell-wise sum $\mathbf{s}_k$ and sum of squares $\mathbf{v}_k$ are calculated using Equations 15 and 16, respectively.

$$\mathbf{s}_k = \sum_{(\mathbf{p}_n, \mathbf{r}_n) \in \mathcal{D}_{\mathrm{tr},k}^N} \mathbf{r}_n \tag{15}$$

$$\mathbf{v}_k = \sum_{(\mathbf{p}_n, \mathbf{r}_n) \in \mathcal{D}_{\mathrm{tr},k}^N} \mathbf{r}_n \odot \mathbf{r}_n \tag{16}$$

Let $N_{\mathrm{tr}} = \sum_{k=1}^K N_k$ denote the total number of normal training samples. The server calculates the cell-wise mean and standard deviation using Equations 17 and 18, respectively.

$$\boldsymbol{\mu}_r = \frac{1}{N_{\mathrm{tr}}} \sum_{k=1}^K \mathbf{s}_k \tag{17}$$

$$\boldsymbol{\sigma}_r = \sqrt{\frac{1}{N_{\text{tr}}}\sum_{k=1}^{K}\mathbf{v}_k - \boldsymbol{\mu}_r \odot \boldsymbol{\mu}_r} \tag{18}$$

In Equation 18, $\odot$ denotes element-wise multiplication. The square root and the other vector operations in Equation 18 are also applied element-wise. Equations 19 and 20 normalize the location and RSRP vectors, respectively, using the service-area metadata and aggregate statistics.

$$\bar{\mathbf{p}}_n = \left(\mathbf{p}_n - \mathbf{c}_p\right) \oslash \mathbf{h}_p \tag{19}$$

$$\bar{\mathbf{r}}_n = \left(\mathbf{r}_n - \boldsymbol{\mu}_r\right) \oslash \boldsymbol{\sigma}_r \tag{20}$$

In Equations 19 and 20, $\oslash$ denotes element-wise division. For numerical stability, a standard deviation sufficiently close to zero is replaced with a predefined positive constant. The normalization parameters are estimated from the normal training partitions and then applied unchanged to the calibration data and all subsequent observations.

3.3 Location-RSRP Consistency-Based FBS Detection

3.3.1 Normal-Only Learning Principle

The proposed system does not train a supervised binary classifier to distinguish FBS attacks from normal conditions. Instead, each regression model learns normal location-RSRP consistency using only normal training data. This normal-only design reflects the availability of data in real-world cellular networks. Because FBS attacks occur infrequently in operational environments, collecting a sufficient number of representative and correctly labeled attack samples is difficult. In contrast, normal location-RSRP measurements can be collected continuously from UEs during ordinary network operation. The proposed detector therefore learns the normal relationship between location and RSRP and identifies observations that violate this relationship under the considered threat model. Equation 21 defines the global normal training set, and Equation 22 defines its size N_{tr}.

$$\mathcal{D}_{\text{tr}}^{N} = \bigcup_{k=1}^{K} \mathcal{D}_{\text{tr},k}^{N} \tag{21}$$

$$N_{\text{tr}} = \sum_{k=1}^{K} \left|\mathcal{D}_{\text{tr},k}^{N}\right| \tag{22}$$

The union in Equation 21 is a mathematical representation of the normal training population across regions. It does not require the server to materialize the union or receive the underlying raw samples.

FBS labels are excluded from regression training and threshold calibration. Instead, a high quantile of anomaly scores from normal calibration data defines the acceptance range. The models therefore learn no attack-specific patterns, and an out-of-range observation indicates FBS-induced location-RSRP inconsistency under the considered threat model.

Forward regression predicts the normal RSRP vector **r** from the UE location **p** and uses the maximum positive RSRP residual in dB as its anomaly score. Inverse regression estimates **p** from **r** and uses the Euclidean location error in meters. These formulations are evaluated as alternatives, and only one is selected for deployment.

3.3.2 Forward RSRP Prediction Detector

Given a normalized UE location, the forward model f_θ predicts the corresponding normal RSRP vector. Equation 23 defines the normalized forward prediction, and Equation 24 converts the prediction to dBm.

$$\bar{\hat{\mathbf{r}}}_n = f_\theta\left(\bar{\mathbf{p}}_n\right) \tag{23}$$

$$\hat{\mathbf{r}}_n = \boldsymbol{\mu}_r + \boldsymbol{\sigma}_r \odot \bar{\hat{\mathbf{r}}}_n \tag{24}$$

The forward model is trained by minimizing the mean squared error over all N_{tr} normal samples and M output components, as shown in Equation 25.

$$\min_{\theta} \frac{1}{MN_{\text{tr}}} \sum_{(\mathbf{p}_n,\mathbf{r}_n)\in\mathcal{D}_{\text{tr}}^{N}} \left\|\bar{\mathbf{r}}_n - f_\theta\left(\bar{\mathbf{p}}_n\right)\right\|_2^2 \tag{25}$$

During inference, Equation 26 defines the forward anomaly score as the largest positive difference between an observed RSRP value and its predicted normal value.

$$s_{\text{fwd}}\left(\mathbf{p}_n,\mathbf{r}_n\right) = \max_{m\in\{1,\ldots,M\}} \left[r_n^{(m)} - \hat{r}_n^{(m)}\right]_+ \tag{26}$$

If at least one observed RSRP value exceeds its normal prediction, the largest positive residual becomes the anomaly score. The detector does not standardize the residuals or calculate an aggregate ℓ_2-norm across the cells. Under the proposed threat model, negative residuals, which represent decreases in RSRP, are set to zero.

3.3.3 Inverse Location Estimation Detector

Given a normalized RSRP vector, the inverse model g_ϕ estimates the UE location. Equation 27 defines the normalized location estimate, and Equation 28 converts the estimate to coordinates in meters.

$$\bar{\hat{\mathbf{p}}}_n = g_\phi\left(\bar{\mathbf{r}}_n\right) \tag{27}$$

$$\hat{\mathbf{p}}_n = \mathbf{c}_p + \mathbf{h}_p \odot \bar{\hat{\mathbf{p}}}_n \tag{28}$$

Because the inverse model produces two coordinate components, it is trained using the mean squared error averaged over both components and all normal training samples, as shown in Equation 29.

$$\min_{\phi} \frac{1}{2N_{\mathrm{tr}}} \sum_{(\mathbf{p}_n,\mathbf{r}_n)\in\mathcal{D}_{\mathrm{tr}}^{N}} \left\| \bar{\mathbf{p}}_n - g_\phi\left(\bar{\mathbf{r}}_n\right) \right\|_2^2 \tag{29}$$

Equation 30 defines the inverse anomaly score as the Euclidean distance between the estimated and actual UE locations.

$$s_{\mathrm{inv}}\left(\mathbf{p}_n,\mathbf{r}_n\right) = \left\| \mathbf{p}_n - \hat{\mathbf{p}}_n \right\|_2 \tag{30}$$

If an FBS changes the observed RSRP pattern to resemble the pattern of another geographic region, the estimated location can move away from the actual location and increase the anomaly score. Although the regression input of the inverse model contains only the RSRP vector, calculation of the anomaly score also requires the actual UE location $\mathbf{p}_n$. Therefore, the inverse approach is not an RSRP-only detector that operates without location information.

3.3.4 Region-Specific Threshold Calibration and Decision

Although all regions use the same global model, their normal score distributions can differ because their radio environments and spatial distributions are non-IID. Client k therefore applies the global model to the held-out normal calibration set of region $\mathcal{R}_k$ and determines a region-specific threshold. For detector $d\in\{\mathrm{fwd}, \mathrm{inv}\}$ and region k, Equation 31 defines the threshold as the q-quantile of the corresponding normal calibration scores.

$$\tau_{d,k} = Q_q\left(\left\{ s_d\left(\mathbf{p}_n,\mathbf{r}_n\right) \middle| \left(\mathbf{p}_n,\mathbf{r}_n\right)\in\mathcal{D}_{\mathrm{cal},k}^{N} \right\}\right) \tag{31}$$

In Equation 31, $Q_q(\cdot)$ denotes the empirical quantile, where $q\in(0,1)$. The threshold is estimated using only the held-out normal calibration set of region $\mathcal{R}_k$, which is not used for model training. For an observation located in that region, Equation 32 assigns $\hat{\ell}_{n,k}^{d}=1$ when the anomaly score exceeds the regional threshold and assigns $\hat{\ell}_{n,k}^{d}=0$ otherwise.

$$\hat{\ell}_{n,k}^{d} = \begin{cases} 1, & s_d\left(\mathbf{p}_n,\mathbf{r}_n\right) > \tau_{d,k} \\ 0, & s_d\left(\mathbf{p}_n,\mathbf{r}_n\right) \le \tau_{d,k} \end{cases} \tag{32}$$

Equation 33 defines the resulting location-RSRP acceptance set for detector d in region $\mathcal{R}_k$.

$$\mathcal{A}_{d,k} = \left\{ (\mathbf{p},\mathbf{r}) \middle|\ s_d\left(\mathbf{p},\mathbf{r}\right) \le \tau_{d,k} \right\} \tag{33}$$

Equation 33 provides the operational definition of location-RSRP consistency in each supported region. Because the forward and inverse anomaly scores have different physical units, a separate regional threshold is calibrated for each candidate. After one formulation is selected, the final detector uses only the regional threshold map and decision rule associated with that formulation. The two decisions are not combined using an AND or OR rule. The quantile parameter q determines the nominal fraction of normal calibration samples accepted by the detector.

3.3.5 Regression Model Requirements

The proposed detection principle does not depend on a specific regression architecture. The forward model maps a normalized two-dimensional location to an M-dimensional RSRP vector, whereas the inverse model maps a normalized M-dimensional RSRP vector to a two-dimensional location. For federated training, all clients use the same differentiable model architecture and parameter structure so that the server can aggregate their updates.

Depending on the candidate detector direction, the model outputs either the RSRP vector or the location coordinates expected under normal conditions, rather than a class probability. A change in the regression architecture does not change the anomaly scores in Equations 26 and 30 or the region-specific decision rule in Equation 32. Once the detector direction is selected, only the corresponding global model, anomaly score, and regional threshold map are used for deployment.

3.4 Federated Training and UE-Side Inference

3.4.1 Algorithm-Agnostic Federated Objective

In federated training, each client minimizes a detector-specific regression objective F_k^d using its normal local data. Let $\mathbf{w}$ denote the model parameters, $\bar{\mathbf{x}}_n^d$ and $\bar{\mathbf{y}}_n^d$ denote the normalized input and target, respectively, and o_d denote the output dimension. The notation $h_{\mathbf{w}}^{d}$ represents the regression model for detector d, where $h_{\mathbf{w}}^{\mathrm{fwd}} = f_\theta$ and $h_{\mathbf{w}}^{\mathrm{inv}} = g_\phi$. For the forward detector, $\left(\bar{\mathbf{x}}^{\mathrm{fwd}},\bar{\mathbf{y}}^{\mathrm{fwd}},o_{\mathrm{fwd}}\right)=\left(\bar{\mathbf{p}},\bar{\mathbf{r}},M\right)$. For the inverse detector, $\left(\bar{\mathbf{x}}^{\mathrm{inv}},\bar{\mathbf{y}}^{\mathrm{inv}},o_{\mathrm{inv}}\right)=\left(\bar{\mathbf{r}},\bar{\mathbf{p}},2\right)$. Using the regional sample count N_k defined in Section 3.2.4, Equation 34 specifies the corresponding mean squared regression objective.

$$F_k^d\left(\mathbf{w}\right) = \frac{1}{o_d N_k} \sum_{(\mathbf{p}_n,\mathbf{r}_n)\in\mathcal{D}_{\mathrm{tr},k}^{N}} \left\| \bar{\mathbf{y}}_n^d - h_{\mathbf{w}}^d\left(\bar{\mathbf{x}}_n^d\right) \right\|_2^2 \tag{34}$$

The proposed detector formulations do not require a specific federated optimization algorithm. Each client calculates a local update by optimizing the objective in Equation 34 according to the selected federated optimization rule. The server then aggregates the local updates according to the same rule to train the global model. The forward and inverse candidate models are

trained independently and do not share parameters or optimizer states. After the detector direction is selected, only the selected global model is retained for deployment.

Each client calculates its local update using only the normal training set $\mathcal{D}_{\mathrm{tr},k}^{N}$. The normal calibration set and FBS samples are not used for gradient calculation or model aggregation.

3.4.2 Global Model and Region-Specific Threshold-Map Deployment

Once detector direction d has been selected and its model has completed T rounds of federated training, the server fixes the global model $\mathbf{w}_d^T$ and the corresponding normalization metadata $\{\mathbf{c}_p, \mathbf{h}_p, \boldsymbol{\mu}_r, \boldsymbol{\sigma}_r\}$. Client k applies this global model to its held-out normal calibration set and calculates $\tau_{d,k}^T$ according to Equation 31. It then uploads only the tuple containing the region identifier, detector identifier, global-model version, and scalar threshold. Raw calibration pairs and individual anomaly scores are not uploaded.

The server constructs the region-specific threshold map in Equation 35 from the entries registered by all supported regions.

$$\mathcal{T}_d^T = \left\{ \left(\mathcal{R}_k, \tau_{d,k}^T \right) \mid k \in \mathcal{K} \right\} \tag{35}$$

The server distributes the global model, normalization metadata, supported-region boundaries, and complete threshold map $\mathcal{T}_d^T$ to every UE. The model and threshold map are versioned together; a threshold calibrated for one global-model version is not applied to another version. Existing UEs and newly deployed UEs follow the same deployment procedure and do not require UE-specific threshold calibration.

If the normal propagation conditions in region $\mathcal{R}_k$ change substantially, the deployed regression model and $\tau_{d,k}^T$ may no longer maintain the intended false-alarm behavior. The regional client then collects new trusted normal samples. Depending on the extent of the change, the server updates the global model through federated retraining and recalibrates the affected regional threshold, or recalibrates only the threshold when the existing model remains valid. The updated model and threshold map are distributed with a consistent version identifier.

3.4.3 Position-Based Threshold Selection and UE-Side Inference

Let $\mathcal{R} = \bigcup_{k \in \mathcal{K}} \mathcal{R}_k$ denote the supported service area. Equation 36 defines the region-assignment function from the supported service area to the client index set. Equation 37 assigns observation n to region k_n according to its UE location.

$$g : \mathcal{R} \rightarrow \mathcal{K} \tag{36}$$

$$g(\mathbf{p}_n) = k_n \Leftrightarrow \mathbf{p}_n \in \mathcal{R}_{k_n} \tag{37}$$

The assignment function $g(\cdot)$ uses the region boundaries contained in the threshold map and returns one deterministic region index, including at region boundaries. The UE retrieves τ_{d,k_n}^T from $\mathcal{T}_d^T$ and applies the decision rule in Equation 32 with $k = k_n$. When the UE moves to another supported region, it changes only the selected threshold entry; the global model and score definition remain unchanged.

For each observation ($\mathbf{p}_n$, $\mathbf{r}_n$), the UE normalizes the input, predicts the expected normal output using the global model, converts the prediction to physical units, and compares the anomaly score s_d with the regional threshold. The forward detector predicts RSRP from location and calculates a residual score, whereas the inverse detector predicts location from RSRP and compares it with the actual UE location.

All UEs in supported regions, including cold-start UEs that did not participate in training or calibration, can detect immediately after receiving the deployment package. A UE outside $\mathcal{R}$ remains unsupported until its region completes the onboarding process in Section 3.1. Raw samples and anomaly scores remain at regional clients; the server receives only model updates, normalization statistics, and one threshold per region, while inference is performed locally.

These updates and metadata may still leak information. Secure aggregation, differential privacy, encryption, and threshold-map confidentiality are outside the scope of this study; therefore, the system reduces raw-data centralization but does not provide a formal privacy guarantee. Here, UE-side denotes a logical deployment in which UEs or their representative edge clients perform training or inference.

4 Experiments

This section evaluates the detection performance and deployment efficiency of the proposed UE-side FBS detection system. The evaluated regression models include MLP, 1D-CNN, 2D-CNN, GNN, Transformer [16-19], and XGBoost [20]. The neural models are trained using FedAvg, FedProx, SCAFFOLD, and FedDyn to examine the effects of dataset size, regression direction, model architecture, and federated optimization. Experimental data containing UE locations and per-cell received power are generated through radio-only NS3 simulations using the 3GPP UMa model [15, 34].

4.1 Dataset and Preprocessing

The dataset is generated through 28-GHz radio-only NS3 simulations using the 3GPP UMa path-loss and channel-condition models. Rather than instantiating the full 3GPP protocol stack, RRC attachment, bearers, or reference-signal mapping, the simulator calculates the per-cell received power in dBm from transmitter power and transmitter–receiver geometry.

The simulation area is divided into a 3×3 grid with one legitimate transmitter at the center of each region. The cells are identified as 100–108, and one UE moves within each region, resulting in nine federated clients.

The environment uses forced line-of-sight conditions with shadowing disabled.

Normal training, normal test, and FBS data are generated in separate runs using base seed 1 and run indices 1, 2, and 3, respectively. Each sample contains the UE coordinates (x, y) and a nine-dimensional RSRP vector ordered by cell identifiers 100–108.

The FBS, identified as cell 900, is placed between gNBs 4 and 5 and impersonates cell 103. The FBS RSRP replaces the legitimate cell-103 measurement only when it is at least as strong. Only samples in which this replacement occurs are labeled as FBS-affected; unchanged samples are excluded from the FBS class. Figure 1 shows the NS3 simulation configuration used to generate the experimental dataset. The points with different colors represent the UE coordinates collected in the nine client regions. Each region contains 100,000 samples, resulting in 900,000 samples in total. The triangles indicate the locations of the legitimate gNBs at the centers of the regions, and the X marker indicates the FBS location between legitimate gNB 4 and gNB 5.

Samples are assigned to nine client datasets according to UE location. Each client's normal data are shuffled with a fixed seed and split into 85% for regression training and 15% for regional threshold calibration. The balanced test set contains equal numbers of normal and FBS-affected samples from each client, resulting in a 1:1 ratio. FBS samples and labels are excluded from training and calibration.

The forward detector maps UE coordinates to a nine-dimensional RSRP vector, whereas the inverse detector performs the opposite mapping. Coordinates are normalized using the service-area center and half-width, and RSRP values are normalized using cell-wise statistics calculated from the normal training data of all clients. These parameters are applied unchanged to the calibration and test sets to prevent information leakage.

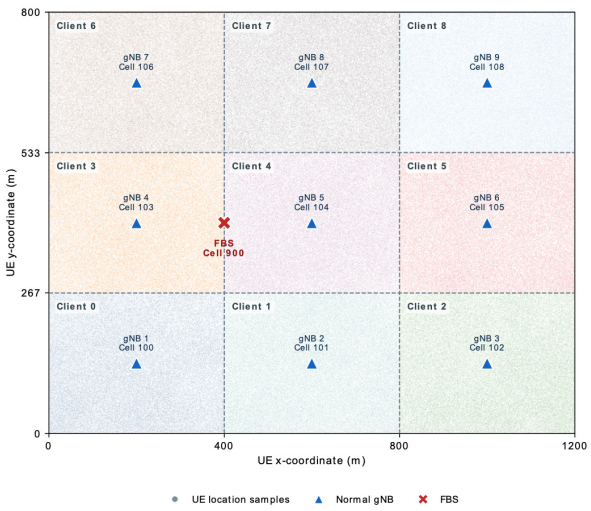

Figure 1. NS3 simulation configuration for dataset generation

Table 1 provides the complete simulation parameters.

Table 1. NS3 simulation parameters used for dataset generation

Parameter	Setting
Area and grid	1200 x 800 m; 3 x 3 regions
Propagation	3GPP UMa; 28 GHz; forced LOS
Legitimate gNBs	9; transmit power: 46 dBm; height: 10 m
gNB coordinates	x = {200, 600, 1000} m; y = {133.33, 400, 666.67} m; nine grid-center combinations
FBS	Source ID: 900; impersonated cell: 103; transmit power: 58 dBm
FBS position	(400, 400, 10) m
UEs	9; one per region; height: 1.5 m
Initial UE position	Uniformly distributed within each assigned region
Mobility	Bounded RandomWalk2d; speed: uniform 1-8 m/s; update interval: 3 s
Sampling interval	10 seconds
Randomization	Base seed: 1; run indices: training 1, normal test 2, and FBS 3

The four dataset configurations and the corresponding balanced test-set sizes are summarized in Table 2.

Table 2. Dataset configurations used in the experiments

Dataset size	Normal samples/ client	Balanced test samples
5k	5,000	88,792
20k	20,000	355,446
50k	50,000	888,550
100k	100,000	1,776,652

The normal sample counts in Table 2 are measured before the 85:15 training-calibration split. The four dataset sizes form nested configurations that use prefixes of the same run trajectory for each data role. Consequently, the numbers of training, calibration, and test samples increase together as the dataset size increases.

4.2 Experimental Environment

All training and inference experiments are performed on the same workstation using the same codebase and a fixed random seed. The hardware and software environment is summarized in Table 3.

Table 3. Hardware and software environment used for training and inference

Component	Setting
Operating System	Ubuntu 24.04 LTS on Windows Subsystem for Linux 2 (WSL2)
CPU	AMD Ryzen 9 9950X3D
RAM	61.62 GiB available to WSL2
GPU	NVIDIA GeForce RTX 5090
Network Simulator	Network Simulator 3.45
Python	3.12.3
Deep learning	PyTorch 2.12.1+cu130
XGBoost	3.3.0
Federated learning	NVIDIA FLARE 2.8.0
CUDA/CuPy	CUDA 13.0, CuPy 14.1.1

Neural federated training is implemented using NVIDIA FLARE [35]. To isolate the effect of model architecture, the neural models are compared under the same training conditions. All models are trained to minimize the mean-squared error (MSE), averaged over all output elements, on the normalized regression targets. The architecture-specific hyperparameters are listed in Table 4.

Table 4. Architecture-specific hyperparameters of the neural models

Model	Hyperparameters
All neural models	Hidden dimension: 384; dropout: 0.02
MLP	Fully connected layers: 4; LayerNorm; SiLU
1D-CNN	Convolutional layers: 3; channels (inverse/forward): [64, 128, 128]/[96, 96, 64]; kernel size: 3; padding: 1; GroupNorm; SiLU
2D-CNN	Convolutional layers: 3; channels (inverse/forward): [32, 64, 96]/[64, 64, 32]; kernel size: 3×3; padding: 1; GroupNorm; SiLU
GNN	Graph convolutional layers: 3; fixed 3×3 four-neighbor graph with self-loops; node dimension: 384; SiLU
Transformer	Encoder layers: 4; attention heads: 8; feed-forward dimension: 1,536; GELU; pre-norm

XGBoost uses the same client partition and calibration and testing protocol as the neural models but is trained using federated bagging in NVIDIA FLARE. The forward detector consists of one booster for each of the nine RSRP targets, whereas the inverse detector consists of one booster for each of the two coordinate targets. The XGBoost hyperparameters are listed in Table 5.

Table 5. Hyperparameters of the XGBoost models

Model	Hyperparameters
XGBoost	Federated mode: bagging; global rounds: 100; local parallel trees per client and round: 1; learning rate: 0.08; maximum depth: 8; subsample: 0.85

The same common federated learning parameters are applied to every neural architecture and both regression directions. Table 6 summarizes the common settings and the algorithm-specific parameters of the four federated learning algorithms.

Table 6. Common and algorithm-specific hyperparameters for neural federated learning

Algorithm	Parameters
Common	Global rounds: 100; clients: 9 (100% participation); local epochs: 1; batch size: 4,096; SGD (learning rate: 0.01; weight decay: 10^{-4})
FedAvg	Sample-count-weighted aggregation
FedProx	μ: 0.01; sample-count-weighted aggregation
SCAFFOLD	Equal client weighting
FedDyn	α: 0.01; maximum gradient norm: 10.0

Table 7 summarizes the model-specific representations. In Table 7, F and I denote the forward and inverse models, respectively. All models use the same location–RSRP pairs, with cells 100–108 ordered row-wise over the 3×3 grid. Let $\mathbf{q}_j \in \mathbb{R}^2$ denote the normalized location of cell j and C the latent channel count.

Table 7. Input representations of the neural models

Model	Input representation
MLP	F: the flat 2-D location $\mathbf{p}$ is mapped to $\mathbf{r}$. I: the flat ordered 9-D RSRP vector $\mathbf{r}$ is mapped to $\mathbf{p}$. Only the fixed cell order is used.
1D-CNN	F: $\mathbf{p}$ is projected to a C×9 latent sequence. I: $\mathbf{r}$ forms a 1×9 sequence. Convolution is applied along the cell-index axis.
2D-CNN	F: $\mathbf{p}$ is projected to a C×3×3 latent grid. I: $\mathbf{r}$ is reshaped to a 1×3×3 cell grid.
GNN	The nine cells are nodes of a four-neighbor graph with self-loops. F: node j uses $[\mathbf{p}, \mathbf{q}_j, \mathbf{p}-\mathbf{q}_j]$ and outputs r_j. I: node j uses $[r_j, \mathbf{q}_j]$, followed by mean and maximum pooling.
Transformer	The nine cells form a length-9 token sequence. F: token j uses $[\mathbf{p}, \mathbf{q}_j, \mathbf{p}-\mathbf{q}_j]$ and outputs r_j. I: token j uses $[r_j, \mathbf{q}_j]$, followed by mean and maximum pooling. No separate positional encoding is used.

Accordingly, the results compare complete architecture–representation combinations under common optimization settings, rather than model capacity alone.

4.3 Result and Analysis

FBS samples are treated as the positive class, and normal samples are treated as the negative class. A true positive (TP) is an FBS sample correctly detected as FBS, whereas a true negative (TN) is a normal sample correctly classified as normal. A false positive (FP) is a normal sample incorrectly detected as FBS, and a false negative (FN) is an FBS sample incorrectly classified as normal. At each round, the region-specific thresholds are calculated by applying $q = 0.99$ to Equation 31 in Section 3.3.4. The TP, TN, FP, and FN counts are then summed across the nine clients to calculate the micro-averaged metrics. Equations 38 and 39 define accuracy and the F1-score, respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{38}$$

$$F1\text{-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{39}$$

The first experiment compares the F1-scores at the end of round 100 on the 100k dataset across detector directions, regression models, and federated learning algorithms. Figure 2 reports the results of applying FedAvg, FedProx, SCAFFOLD, and FedDyn to the neural models, together with the federated bagging results of XGBoost.

Figure 2. Round-100 F1-score heatmaps by regression model and federated learning algorithm on the 100k dataset: (a) forward detector and (b) inverse detector (A dash denotes an inapplicable combination.)

FedDyn achieved the highest forward F1-score for all five neural architectures. Transformer-FedDyn performed best at 0.993, followed by 1D-CNN-FedDyn at 0.991, whereas forward XGBoost achieved 0.928. The choice of federated algorithm had a substantial effect; for example, 1D-CNN achieved approximately 0.855 with FedAvg and FedProx but 0.991 with FedDyn. For the inverse detector, 1D-CNN-SCAFFOLD achieved the highest F1-score of 0.987, followed by 2D-CNN-SCAFFOLD at 0.984 and MLP-FedAvg at 0.982, whereas XGBoost achieved 0.626. The best forward configuration exceeded the best inverse configuration by 0.006, demonstrating that detector direction and federated optimization should be selected jointly.

Selecting a different federated learning algorithm post hoc at each dataset size could overstate the effect of increasing the dataset size. Therefore, for each model and detector direction, Figure 3 first selects the federated learning algorithm with the highest F1-score at 100k and then fixes that algorithm across the 5k, 20k, 50k, and 100k datasets. When two algorithms appear in a legend entry, the first and second algorithms correspond to the forward and inverse detectors, respectively.

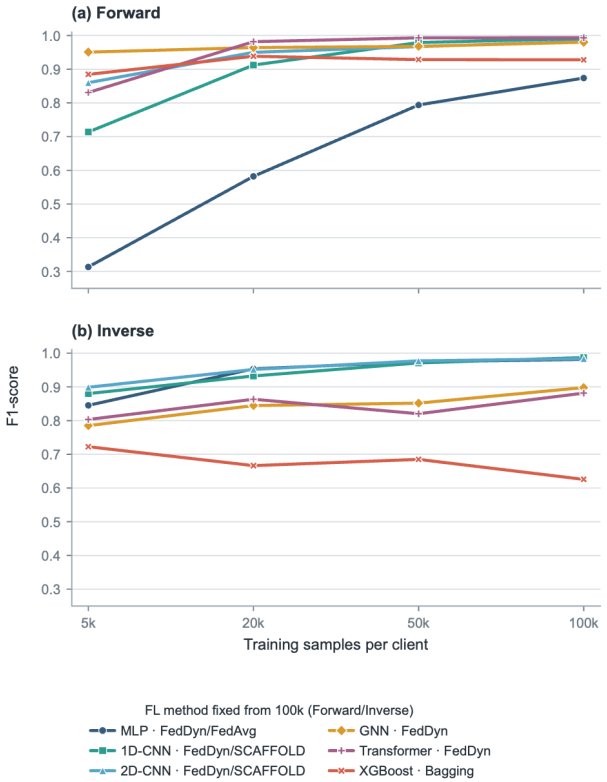

Figure 3. F1-score as a function of dataset size using the fixed federated learning algorithm selected from the 100k results: (a) forward detector and (b) inverse detector

Most neural models improved as the dataset size increased. Forward Transformer-FedDyn increased from an F1-score of 0.831 at 5k to 0.993 at 100k, while forward 1D-CNN-FedDyn increased from 0.714 to 0.991. MLP-FedDyn showed stronger data dependence, improving from 0.313 to 0.874. In contrast, inverse XGBoost decreased

from 0.723 to 0.626. Across all 40 neural configurations, the mean final F1-score increased from 0.561 at 5k to 0.915 at 100k. FedDyn achieved the highest mean F1-score of 0.895, followed by SCAFFOLD at 0.760, FedAvg at 0.713, and FedProx at 0.713. These results show that the benefit of additional data depends on both model architecture and federated optimization.

Figure 4 compares a local-only model trained independently by each client with the global model produced through federated learning. Both models are evaluated on the same client test sets, and the figure reports the difference in the micro-averaged F1-score, defined as $\Delta F1 = F1_{FL} - F1_{local}$. A positive value favors federated learning, whereas a negative value favors local-only training. To maintain a consistent comparison, FedDyn is used for the federated neural models, and federated bagging is used for XGBoost.

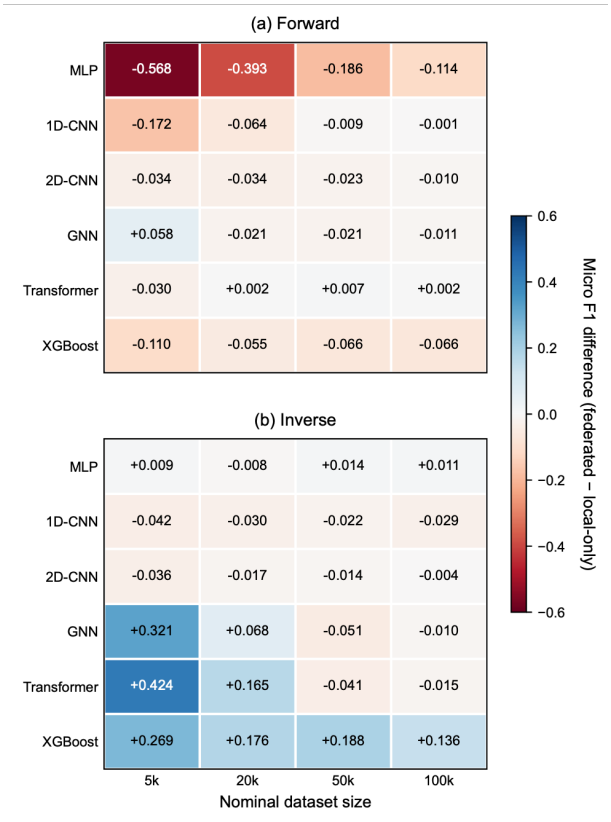

Figure 4. Difference in micro-averaged F1-score between the federated and local-only models by dataset size, detector direction, and model: (a) forward detector and (b) inverse detector (Blue favors federated learning, whereas red favors local-only training.)

Federated learning did not consistently outperform local-only training. For the forward detector, it performed better in only four of 24 comparisons, with a mean ΔF1 of -0.080. The largest local-only advantage occurred for MLP at 5k (ΔF1 = −0.568), because the global model must represent the non-IID patterns of all nine regions rather than the more homogeneous relationship of one region. For the inverse detector, federated learning performed better in 11 comparisons and achieved a mean ΔF1 of 0.061. However, its median was -0.006, indicating that the positive mean was driven by several large improvements under low-data conditions. Differences generally decreased at 100k. Thus, the benefit of federated learning depends on dataset size, detector direction, and model architecture. It also provides a single model spanning multiple regions without centralizing raw location-RSRP samples.

The fourth experiment evaluates the change in detection performance over federated learning rounds. For each regression model, one configuration is selected from all combinations of the forward and inverse detectors and their applicable federated learning algorithms according to the highest F1-score at round 100. Figure 5 compares the round-wise F1-scores of the six selected configurations in a single plot. The light curves show the raw values saved at each round, and the bold curves show five-round moving averages. The legend reports the selected detector direction and federated learning algorithm.

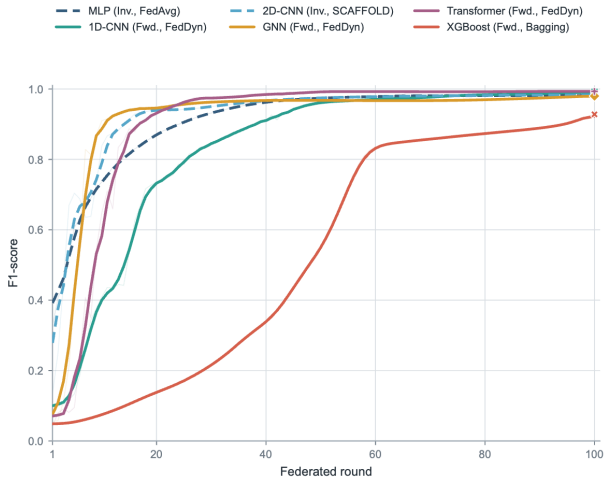

Figure 5. Round-wise F1-score on the 100k dataset after selecting, for each regression model, the detector direction and federated learning algorithm combination with the highest F1-score at round 100

Each model was paired with the detector direction and federated algorithm that produced its highest round-100 F1-score, and this configuration remained fixed throughout the comparison. GNN, 2D-CNN, Transformer, MLP, and 1D-CNN first exceeded an F1-score of 0.90 at rounds 10, 15, 17, 25, and 39, respectively, whereas XGBoost did so at round 95. Transformer achieved the highest final F1-score of 0.993, followed by 1D-CNN at 0.991 despite its slower convergence. XGBoost converged most slowly and produced the lowest final F1-score of 0.928. These results show that convergence speed and final detection performance should be considered separately.

A UE-side detector must account for model deployment size and inference cost in addition to detection performance. Figure 6 compares the best final F1-score of each model at 100k with its model artifact size and CUDA inference throughput on the balanced test set of 1,776,652 samples. Marker area represents artifact size, and points farther to the right indicate greater throughput.

Forward Transformer-FedDyn achieved the highest F1-score of 0.993 but required a 27.108 MiB artifact and 7.292 s to process the balanced test set. Forward 1D-CNN-FedDyn achieved a comparable F1-score of 0.991 with a 1.109 MiB artifact and an inference time of 0.207 s. Thus, its F1-score was only 0.22 percentage points lower, while its artifact was 24.44 times smaller and its throughput was 35.19 times higher. Forward 2D-CNN-FedDyn provided the smallest artifact at 0.720 MiB. MLP achieved the highest inference throughput, although its F1-score was lower at 0.874. In contrast, XGBoost showed an unfavorable deployment tradeoff, with a 133.8 MiB artifact and the lowest inference throughput. These RTX 5090 measurements represent core inference and do not directly indicate latency, memory usage, or energy consumption on an operational UE.

Figure 6. Tradeoff among F1-score, CUDA inference throughput, and model artifact size on the 100k balanced test set: (a) forward detector and (b) inverse detector

In summary, forward Transformer-FedDyn is the preferred configuration when F1-score is the primary criterion, whereas forward 1D-CNN-FedDyn provides the most favorable balance between detection performance and computational efficiency for UE deployment. Forward 1D-CNN-FedDyn is therefore selected as the final deployment candidate. This configuration uses the forward anomaly score in Equation 26 and the region-specific threshold map calibrated for the forward detector; it does not use the inverse detector or combine the two anomaly scores.

4.4 Limitations

The proposed framework improves raw-data locality but does not provide a formal privacy guarantee. Raw training and calibration samples remain at regional clients, whereas model updates, normalization statistics, regional thresholds, and deployment metadata are shared. Model updates may be vulnerable to gradient- or update-inversion attacks, while the aggregate statistics and threshold map may disclose regional data characteristics. Secure aggregation, differential privacy, and dedicated confidentiality protection for model updates and the threshold map are not implemented; poisoning and Byzantine attacks are also not evaluated.

The simulated inputs are propagation-model received powers used as proxies for per-cell RSRP, rather than standards-compliant SS-RSRP measurements. The radio-only generator omits reference-signal resource mapping, standardized RSRP reporting quantization, Layer-1/Layer-3 filtering, and FR2 beamforming and beam management. These omissions may affect anomaly-score distributions, calibrated thresholds, and detection performance. In addition, Equations 9–12 select the stronger dBm value for an impersonated cell rather than modeling waveform-level interactions. Depending on synchronization and receiver processing, a physical receiver may instead experience power combination, interference, capture, or decoding failure.

The evaluation uses nine fully participating clients in a fixed 3 × 3 grid, LOS propagation without shadowing, and one fixed FBS location, transmit power, and impersonated cell. The RTX 5090 measurements cover core model inference only and do not represent end-to-end UE latency, runtime memory, energy consumption, or on-device performance. Larger deployments, partial client participation, and federated communication overhead remain to be evaluated.

5 Conclusion

The proposed system addresses two practical challenges in FBS detection: the scarcity of representative attack samples with reliable labels and the privacy risk of centrally collecting UE locations and RSRP measurements. It learns location-RSRP consistency exclusively from normal data and detects FBS-induced mismatches between UE locations and observed RSRP patterns. Federated learning allows geographically separated non-IID clients to train a global regression model without sharing raw measurements. The global model and region-specific threshold map are then distributed to UEs for local detection.

Federated learning did not consistently improve detection performance over local-only training. It outperformed local-only models in only 4 of 24 forward comparisons (mean $\Delta F1 = -0.080$) and 11 of 24 inverse comparisons (mean $\Delta F1 = 0.061$); moreover, the positive mean for the inverse detector was driven mainly by several large gains under low-data conditions. These

results indicate that a global model can incur a detection-performance cost when representing heterogeneous regional data. The principal value of federation is therefore operational rather than a universal improvement in F1-score: it can support data-scarce regions, provide a single versioned model-and-threshold deployment package, and enable cold-start UEs to perform detection immediately within trained and calibrated regions without centrally collecting raw measurements.

Increasing the dataset size from 5k to 100k samples per client raised the average final F1-score across 40 neural configurations from 0.561 to 0.915. FedDyn achieved the highest average F1-score of 0.895 among the evaluated federated algorithms. Forward Transformer-FedDyn provided the best overall F1-score of 0.993, and the selected neural configuration outperformed XGBoost for every dataset size and detection direction. Forward 1D-CNN-FedDyn achieved an F1-score only 0.002 lower than Transformer-FedDyn while using a 1.109 MiB model artifact that was 24.44 times smaller and providing 35.19 times higher inference throughput. It therefore offered the most favorable balance between detection performance and deployment efficiency. These results show that dataset size, regression direction, model architecture, and federated optimization must be considered jointly.

Future work will evaluate the system beyond the radio-only Network Simulator 3 environment under NLOS propagation, shadowing, UE mobility, network reconfiguration, and diverse FBS configurations. PHY-aware signal modeling that accounts for synchronization, interference, signal superposition, and decoding outcomes will be incorporated to assess the validity of the max-selection abstraction under more realistic radio conditions. Operational UE measurements will also be used to assess robustness in real-world radio environments. Continual federated learning, regional recalibration, domain adaptation, and on-device implementation will be investigated. Further evaluation will address scalability, communication overhead, inference latency, energy consumption, and robustness against poisoning and Byzantine attacks. Secure aggregation and differential privacy will also be investigated to provide stronger privacy protection, together with their effects on detection performance, communication overhead, and computational cost.

References

[1] L. Arnold, M. Hollick, J. Classen, Catch You Cause I Can: Busting Rogue Base Stations Using CellGuard and the Apple Cell Location Database, *27th International Symposium on Research in Attacks, Intrusions and Defenses*, Padua, Italy, 2024, pp. 613–629. https://doi.org/10.1145/3678890.3678898

[2] L. Karaçay, Z. Bilgin, A. B. Gündüz, P. Çomak, E. Tomur, E. U. Soykan, U. Gülen, F. Karakoç, A Network-Based Positioning Method to Locate False Base Stations, *IEEE Access*, Vol. 9, pp. 111368–111382, August, 2021. https://doi.org/10.1109/ACCESS.2021.3103673

[3] P. K. Nakarmi, J. Sternby, I. Ullah, Applying Machine Learning on RSRP-Based Features for False Base Station Detection, *17th International Conference on Availability, Reliability and Security*, Vienna, Austria, 2022, pp. 1–7. https://doi.org/10.1145/3538969.3543787

[4] Z. Li, W. Wang, C. Wilson, J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, Y. Liu, FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild, *24th Annual Network and Distributed System Security Symposium*, San Diego, CA, 2017, pp. 1–15. https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017_08-3_Li_paper.pdf

[5] I. W. A. J. Pawana, H. Kwon, Linawati, I. You, Quantum-Enhanced Detection of False Base Stations in Beyond 5G Networks Using Quanvolutional Neural Network, *2025 International Conference on Smart-Green Technology in Electrical and Information Systems*, Denpasar, Bali, Indonesia, 2025, pp. 207–212. https://doi.org/10.1109/ICSGTEIS68532.2025.11284521

[6] T. Heijligenberg, D. Rupprecht, K. Kohls, The Attacks Aren't Alright: Large-Scale Simulation of Fake Base Station Attacks and Detections, *17th Cyber Security Experimentation and Test Workshop*, Philadelphia, PA, 2024, pp. 54–64. https://doi.org/10.1145/3675741.3675742

[7] S. Park, *Why We Cannot Win: On Fake Base Stations and Their Detection Methods*, Ph. D. Thesis, Technische Universität Berlin, Berlin, Germany, 2023. https://doi.org/10.14279/depositonce-18295

[8] D. Son, Y. Park, B. Kim, I. You, A Study on the Implementation of a Network Function for Real-Time False Base Station Detection for the Next Generation Mobile Communication Environment, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 15, No. 1, pp. 184–201, March, 2024. http://doi.org/10.58346/JOWUA.2024.I1.013

[9] K.-W. Huang, H.-M. Wang, Identifying the Fake Base Station: A Location Based Approach, *IEEE Communications Letters*, Vol. 22, No. 8, pp. 1604–1607, August, 2018. https://doi.org/10.1109/LCOMM.2018.2843334

[10] P. Ney, I. Smith, G. Cadamuro, T. Kohno, SeaGlass: Enabling City-Wide IMSI-Catcher Detection, *Proceedings on Privacy Enhancing Technologies*, Vol. 2017, No. 3, pp. 39–56, July, 2017. https://doi.org/10.1515/popets-2017-0027

[11] P. K. Nakarmi, M. A. Ersoy, E. U. Soykan, K. Norrman, Murat: Multi-RAT False Base Station Detector, *arXiv preprint*, arXiv:2102.08780, February, 2021. https://arxiv.org/abs/2102.08780

[12] H. Park, P. V. B. Astillo, Y. Ko, Y. Park, T. Kim, I. You, SMDFbs: Specification-Based Misbehavior Detection for False Base Stations, *Sensors*, Vol. 23, No. 23, Article No. 9504, December, 2023. https://doi.org/10.3390/s23239504

[13] H. Park, P. V. B. Astillo, T. Kim, I. You, 5G Native Network Function for False Base Station Detection Using Machine Learning Technique, *Information Systems Frontiers*, Vol. 27, No. 6, pp. 2279–2294, December, 2025. https://doi.org/10.1007/s10796-025-10614-1

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, *20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, 2017, pp. 1273–1282.

[15] The ns-3 Project, *ns-3 Model Library*, Release ns-3.45, June, 2025. https://www.nsnam.org/releases/ns-3-45/

[16] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning

Representations by Back-Propagating Errors, *Nature*, Vol. 323, No. 6088, pp. 533–536, October, 1986. https://doi.org/10.1038/323533a0

[17] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, November, 1998. https://doi.org/10.1109/5.726791

[18] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, *5th International Conference on Learning Representations*, Toulon, France, 2017, pp. 1–14. https://openreview.net/forum?id=SJU4ayYgl

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, *31st Conference on Neural Information Processing Systems*, Long Beach, CA, 2017, pp. 5998–6008. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[20] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated Optimization in Heterogeneous Networks, *Third Conference on Machine Learning and Systems*, Austin, TX, 2020, pp. 429–450.

[22] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, A. T. Suresh, SCAFFOLD: Stochastic Controlled Averaging for Federated Learning, *37th International Conference on Machine Learning*, Virtual Conference, 2020, pp. 5132–5143.

[23] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, V. Saligrama, Federated Learning Based on Dynamic Regularization, *9th International Conference on Learning Representations*, Virtual Conference, 2021, pp. 1–36. https://openreview.net/forum?id=B7v4QMR6Z9w

[24] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, E. Weippl, IMSI-Catch Me If You Can: IMSI-Catcher-Catchers, *30th Annual Computer Security Applications Conference*, New Orleans, LA, 2014, pp. 246–255. https://doi.org/10.1145/2664243.2664272

[25] Int64, *EAGLE Security*, Android application, 2014.

[26] C. Quintin, Detecting Fake 4G LTE Base Stations in Real Time, *USENIX Enigma 2021*, Virtual Conference, 2021.

[27] K. S. Mubasshir, I. Karim, E. Bertino, Gotta Detect 'Em All: Fake Base Station and Multi-Step Attack Detection in Cellular Networks, *34th USENIX Conference on Security Symposium*, Seattle, WA, 2025, pp. 5365–5384.

[28] V. Abella, I. W. A. J. Pawana, I. You, H. Park, MODI: On-Device Preprocessing of Qualcomm Diagnostic Logs for Cellular Protocol Analysis on Android, *Journal of Internet Services and Information Security*, Vol. 16, No. 1, pp. 612–623, February, 2026. https://doi.org/10.58346/JISIS.2026.I1.034

[29] Y. Park, I. W. A. J. Pawana, B. Kim, I. You, Harnessing GAN to Create Realistic FBS(False Base Station) Attack Data in 5G, *2024 IEEE International Symposium on Consumer Technology*, Kuta, Bali, Indonesia, 2024, pp. 404–408. https://doi.org/10.1109/ISCT62336.2024.10791271

[30] E. S. Briggs, Z. Ji, *Detection of a Rogue Base Station*, U.S. Patent 10,129,283, November 13, 2018.

[31] I. A. Khan, I. Razzak, D. Pi, N. Khan, Y. Hussain, B. Li, T. Kousar, Fed-Inforce-Fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks, *Information Fusion*, Vol. 101, Article No. 102002, January, 2024. https://doi.org/10.1016/j.inffus.2023.102002

[32] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How To Backdoor Federated Learning, *23rd International Conference on Artificial Intelligence and Statistics*, Virtual Conference, 2020, pp. 2938–2948.

[33] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent, *31st Conference on Neural Information Processing Systems*, Long Beach, CA, 2017, pp. 119–129. https://papers.nips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[34] 3rd Generation Partnership Project, *Study on Channel Model for Frequencies from 0.5 to 100 GHz*, Technical Report No. 38.901, Version 18.0.0, Release 18, May, 2024.

[35] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, Z. Zhang, W. Li, A. Myronenko, D. Yang, S. Yang, N. Rieke, A. Quraini, C. Chen, D. Xu, N. Ma, P. Dogra, M. Flores, A. Feng, NVIDIA FLARE: Federated Learning from Simulation to Real-World, *IEEE Data Engineering Bulletin*, Vol. 46, No. 1, pp. 170–184, March, 2023.

Biographies

Hoonyong Park received the B.S. and Ph.D. degrees in Information Security Engineering from Soonchunhyang University, Asan, South Korea, in 2019 and 2023, respectively. He is currently a researcher at AUTOCRYPT Co., Ltd., Seoul, South Korea.

I Wayan Adi Juliawan Pawana (Member, IEEE) received the S.Kom. and M.T. degrees from Udayana University, Indonesia, in 2014 and 2021, respectively. He is currently a Ph.D. student in Cyber Security at Kookmin University, South Korea. His research interests include 5G/6G networks, internet security, and AI security.

Ilsun You (Senior Member, IEEE) received Ph.D. degrees from Dankook University, South Korea, and Kyushu University, Japan. He is a Full Professor of Cyber Security at Kookmin University, South Korea. His research interests include internet security, authentication, access control, and formal security analysis. He is a Fellow of IET.