EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

LPWAN Localization via RSSI/SNR Fingerprinting and Lightweight Machine Learning

Désiré Guel[1,*], Flavien Hervé Somda[1], P. Justin Kouraogo[1], Boureima Zerbo[2], Oumarou Sié[3]

[1]Université Joseph KI-ZERBO (U-JKZ), Burkina Faso
[2]Université Thomas SANKARA (UTS), Burkina Faso
[3]Université Aube Nouvelle, Ouagadougou, Burkina Faso

Abstract

Accurate geolocation for low-power wide-area (LPWAN) devices is desirable when GNSS is unavailable or too energy-expensive, yet RSSI-/TDoA-based approaches are often fragile under channel variability, collisions and cross-device heterogeneity. We address this gap with a reproducible, tabular pipeline that maps LoRa RSSI/SNR/ToA and PHY metadata to 2D positions, compares strong tabular baselines (k-NN, Random Forest, LightGBM, XGBoost), and crucially evaluates them under group-aware (device-wise) splits to avoid identity leakage. On an ns-3-generated LoRa dataset of about 3.3×10^4 labeled receptions, Random Forest attains the tightest distribution with p50 ≈ 0 m and p95 < 1 m, whereas k-NN, despite a low median, exhibits a much heavier tail (p95 ≈ 187 m), underscoring the need to report both central and tail metrics. These results indicate that simple, edge-feasible models can perform gateway-side inference with robust accuracy when fed cleaned features and evaluated with realistic splits, making the approach attractive for practical LPWAN/IoT deployments.

Received on 12 November 2025; accepted on 12 May 2026; published on 01 June 2026

Keywords: LPWAN, LoRa, Localization, Machine Learning, Fingerprinting, RSSI, SNR, IoT

doi:10.4108/eetiot.10888

1. Introduction

Accurate, low-power localization is essential for dense IoT deployments, yet GNSS degrades indoors due to multipath and attenuation [1–3]. LPWANs—notably LoRa/LoRaWAN—offer long range and ultra-low energy, but converting PHY indicators (RSSI/SNR) into precise positions is challenging because of non-stationary channels, device heterogeneity and heavy-tailed errors [4–8].

Classical geometric methods (trilateration, TDoA/AoA) and data-driven fingerprinting have been explored across Wi-Fi/BLE/UWB/LoRa [1, 9–11]. Learning-based fingerprinting alleviates explicit channel modeling but can struggle to generalize across devices and sites without robust protocols and features [2, 12]. Recent LoRa studies report gains via ML/DL on RSSI/SNR [5, 6, 13], dynamic path-loss modeling [14], noise-aware fingerprint cleaning [15] and modern CNN/SE blocks [16]; data-centric augmentation further reduces collection burdens [17]. Community datasets (e.g., multi-storey RSSI/SNR) enable fairer comparisons [18].

We target practical LoRa localization with models that are easy to train, efficient at in ference an d evaluated under group-aware splits (e.g., by device/gateway) with geodesic metrics (median, p90/p95/p99) advocated by prior work [1, 2, 8]. Unlike studies that emphasize novel neural architectures, we position this work as a reproducible benchmark of strong tabular methods, with a focus on fair training, preprocessing and leakage-aware evaluation. We study robustness via ablations and noise handling [15, 19, 20].

Contributions:

- Reproducible pipeline. End-to-end LoRa RSSI/SNR → (lat,lon) with standardized preprocessing, group-aware splits and geodesic reporting [1, 2, 4, 5].

*Corresponding author. Email: desire.guel@ujkz.bf

- Fair baseline benchmark. Head-to-head comparison of tabular localization models (k-NN, Random Forest, LightGBM, XGBoost) under realistic, leakage-aware evaluation.
- Practical insights. Empirical analysis of median vs tail error, cross-device generalization, and gateway-side feasibility in LPWAN/IoT settings.

Research Questions (RQs):

- RQ1: Under group-aware evaluation, which lightweight tabular model best balances central accuracy and tail robustness for LoRa RSSI/SNR fingerprinting? [8, 13]
- RQ2: Which features (RSSI, SNR, path-loss proxies, device/gateway, temporal) most reduce median and tail errors and how sensitive are gains to noise handling [14, 15, 19]?
- RQ3: How well do models generalize across devices/regions and low-data regimes and what benefits arise from channel-aware augmentation [4, 17]?

2. Background and Related Work

This section briefly reviews the main LPWAN localization paradigms, their key challenges, and the role of deep learning in improving localization performance.

2.1. LPWAN localization paradigms.

In LoRa/LoRaWAN and related LPWANs, three families dominate: (i) fingerprinting that maps received indicators (RSSI/SNR/CSI and metadata) to position using ML/DL [8, 12, 13, 21, 22]; (ii) model-based methods that calibrate or adapt path-loss/channel models (possibly with dynamic or context-aware parameters) and invert range estimates [14, 23, 24]; and (iii) time-based, multi-gateway methods (ToA/TDoA/AoA) that solve geometric constraints-typically hyperbolic-when synchronization and densified infrastructure are available [10, 11, 25]. Surveys across indoor positioning consistently situate these families in terms of infrastructure cost, energy and achievable accuracy [1–3]. For LoRaWAN specifically, recent studies show that RSSI/SNR fingerprinting and improved path-loss modeling can be competitive under practical deployments, while TDoA benefits from gateway timing quality and topology [4, 5, 14].

2.2. Challenges in LPWAN localization.

Indoor and mixed environments induce multipath, shadowing and heavy-tailed errors that confound ranging assumptions and degrade naive fingerprints [2, 7, 26]. Hardware bias/device heterogeneity (front-ends, antennas, firmware) shifts RSSI/SNR distributions across devices [27, 28], while ADR (Adaptive Data Rate), spreading factor/coding-rate changes and regional duty-cycle limits alter link budgets and spatiotemporal sampling density [5, 29, 30]. Data quality issues (outliers, missing receptions) further motivate noise-robust modeling and cleaning [15].

2.3. Deep learning (DL) for RF localization: promise and pitfalls.

DL can capture non-linear propagation and cross-feature interactions, improving accuracy over classical regressors in RSSI/SNR fingerprinting [13, 16, 31, 32]. However, overfitting to device/site-specific artefacts and optimistic random splits are recurrent risks; robust evaluation demands group-aware splits (e.g., by device/gateway/region) and clear reporting of median and tail errors (p90/p95) with CDFs [1, 2, 8, 14, 33]. Recent works address data scarcity and drift via channel-aware augmentation and noise detection, improving generalization at lower collection cost [15, 17, 19]. Within LoRa/LPWAN, the balance between lightweight models (for edge feasibility) and robustness to ADR/heterogeneity remains an active design trade-off [4, 29]. Table 1 summarizes the main LPWAN localization paradigms, their requirements, and their main strengths and limitations in LoRa/LoRaWAN settings.

Unlike many prior LPWAN localization studies that report only mean or median error on random splits, this work emphasizes leakage-aware evaluation by grouping samples by device and context. In contrast to research that focuses on complex neural architectures, we show that strong tabular baselines are valuable reproducible references for LoRa RSSI/SNR fingerprinting and that tail metrics are decisive for deployment. Table 2 provides a compact taxonomy of LPWAN localization approaches together with their main challenges and recommended reporting practices.

3. Dataset and Preprocessing

This section describes the simulated LoRa/LPWAN dataset, the main preprocessing steps, and the engineered features used for model training and evaluation.

3.1. Signals and metadata

We train and evaluate on a LoRa/LPWAN dataset exported by the provided `ns-3` program, which logs every successfully received packet at the gateway. In the experiment reported here, the simulator places *one* gateway at the origin $(0, 0)$ and uniformly scatters end devices in a square area of $1{,}000 \times 1{,}000$ m; 50 simulated devices generate in total 32,692

Table 1. LPWAN localization paradigms: requirements, pros/cons and LoRa/LoRaWAN considerations.

Paradigm	Key requirements	Strengths	Limitations / LoRa-specific notes
Fingerprinting (RSSI/SNR/CSI) [8, 12, 13]	Labeled radio map; consistent device/gateway metadata; periodic refresh	No strict sync; leverages multipath; ML/DL captures non-linearities	Sensitive to drift, device bias, ADR changes; needs outlier/noise handling [15, 27]; update cost
Model-based (Path-loss / hybrid) [14, 23, 24]	Calibration (path-loss exponent, offsets); environmental context	Interpretable; less labeling; can fuse with filtering (KF)	Indoor non-stationarity; wall/material effects; exponent/context must adapt [14]
TDoA / AoA (multi-gateway) [10, 11, 25]	Dense gateways; time/phase calibration; backhaul	Strong geometry when sync is good; GNSS-free TDoA variants possible	Gateway sync/jitter; infrastructure cost; duty-cycle/backhaul constraints in LoRaWAN [4, 5]

Table 2. Taxonomy of LPWAN localization approaches (LoRa/LoRaWAN), with challenges and reporting practices

Category	Core Approach	Typical methods / notes	References
Fingerprinting (RSSI/SNR/CSI)	Fingerprint maps (radio signatures)	Room-/area-level or sub-meter with dense anchors and stable RF; CSI improves robustness vs. pure RSSI	[8, 12, 13]
	Refinements (ML/DL; data hygiene)	KNN/RF, MLP/CNN/RNN; noise cleaning, outlier removal and data augmentation to reduce drift	[15, 17]
Model–based	Path–loss / hybrid models	Log–distance, adaptive/hybrid with local calibration; complements sparse fingerprints	[14, 23]
	Refinements (dynamic & filtering)	Dynamic path–loss exponents; Kalman filtering to mitigate multipath/noise	[14]
Time–based	TDoA / AoA	Time/angle geometry with multiple gateways; sensitive to sync and clock offset	[10, 11]
	Refinements (solvers & sync)	Hyperbolic solvers; robustness to timestamp outliers and gateway drift	[10]
Challenges: Multipath/shadowing [7, 26]; hardware bias [27]; ADR/SF and duty–cycle constraints [5, 29]; data drift/noise [15].			
Reporting: Report median/p90/p95 and CDFs; use group–aware splits to avoid leakage [1, 2, 8, 33].			

receptions (after filtering) toward the gateway. The PHY parameters are drawn from EU868–like pools (e.g., SF $\in \{7, \ldots, 12\}$, BW $\in \{125, 250\}$ kHz, $f \in [868, 869]$ MHz) with ADR enabled, so that the exported CSV reflects realistic LoRaWAN time–on–air (ToA) and link–budget variations as in [5, 29].

The CSV header is:

```
DeviceID, Latitude, Longitude,
Distance, TxPower_dBm, SF, BW_Hz,
Freq_Hz, ToA_ms, RSSI_dBm, SNR_dB,
PathLoss_dB, ADR, Start_s, Success,
Colliders
```

In this simulator, `Latitude` and `Longitude` are *planar*, simulator-frame coordinates in meters not geodetic degrees. To make this explicit in the learning code, we rename them to

$$\texttt{x\_m} := \texttt{Latitude}, \qquad \texttt{y\_m} := \texttt{Longitude},$$

and we evaluate localization with the Euclidean error $\|(\hat{x}, \hat{y}) - (x, y)\|_2$. When working with real LoRaWAN traces (geodetic lat/lon), these (x_m, y_m) must be mapped to a local ENU/UTM frame or to a fixed geodetic origin before computing haversine/geodesic errors, as recommended by [1, 2]. We also retain:

- core RF indicators: `rssi_dbm` ≡ `RSSI_dBm`, `snr_db` ≡ `SNR_dB`, `txpower_dbm` ≡ `TxPower_dBm`;
- PHY/context: `sf` ≡ `SF`, `bw_hz` ≡ `BW_Hz`, `freq_hz` ≡ `Freq_Hz`, `toa_ms` ≡ `ToA_ms`;
- ADR flag: `ADR` $\in \{0, 1\}$;

- temporal index: `Start_s` (simulation time of the event);
- reception quality: Success $\in \{0, 1\}$;
- collision report: `Colliders` (list/count of concurrent transmissions on the same time/frequency/SF).

For modeling we derive a normalized boolean `colli-sion_prox` feature such that

$$\texttt{collision\_prox} = \begin{cases} 1 & \text{if } \texttt{Colliders} > 0, \\ 0 & \text{otherwise,} \end{cases}$$

which captures collision/capture context in a form usable by tree and DL models, following the noise–aware fingerprint processing in [15].

3.2. Schema and usage

Table 3 maps the CSV columns to types, units and downstream uses. The selected features match what is commonly reported in RSSI/SNR fingerprinting, path–loss based localization and LoRaWAN measurement studies [5, 8, 14, 28].

3.3. Cleaning and sanity checks

Radio indicators in LoRa/LPWAN are known to be heavy-tailed due to multipath, log-normal shadowing, device-front-end bias and protocol dynamics such as ADR and SF/CR changes [4, 7, 27, 29]. To prevent these effects from dominating the regressors, we apply conservative, literature-backed filters:

- Row validity. Drop any row with missing $\{x_m, y_m\}$ or core RF/PHY features (`RSSI_dBm`, `SNR_dB`, `SF`, `BW_Hz`, `Freq_Hz`); drop Success = 0 since failed decodes do not provide reliable fingerprints. This follows the supervised setup in [1, 2].
- Duplicate removal. Keep only the first occurrence per ⟨`DeviceID`, `Start_s`⟩ to avoid over-counting temporally correlated samples, as recommended in indoor/RSSI evaluations [33].
- Physical plausibility. We bound RSSI to [−130, −40] dBm and SNR to [−25, 25] dB, which covers typical LoRaWAN measurement ranges reported in [5, 7, 28]; samples outside these ranges are flagged and dropped if they are also inconsistent with `PathLoss_dB`. We also verify that `ToA_ms` is consistent with the LoRa ToA model for the chosen SF/BW/payload, as in [14].
- Collision-aware cleaning. If `Colliders` > 0 and Success = 0, the sample is removed (interference-dominated). If `Colliders` > 0 but Success = 1, the row is kept and `collision_prox`= 1 is added, following noise-handling practices from [15].

These thresholds are deliberately slightly wider than the ranges reported in field LoRaWAN studies [5, 28] to accommodate simulator randomness, while still removing values that would give unrealistically optimistic or pessimistic localization errors.

3.4. Feature engineering

We reuse the usual RSSI/SNR fingerprinting features and add a small number of context cues:

- Numeric: `RSSI_dBm`, `SNR_dB`, `TxPower_dBm`, `ToA_ms`, `Freq_Hz` and when present `PathLoss_dB`, standardized on the *training* partition only.
- Categorical PHY: one-hot encodings of `SF`, `BW_Hz` and `ADR`.
- Temporal: from `Start_s` we derive periodic features (e.g., hour-of-day as sin/cos) to let the model capture load-/time-related drift, as suggested in [2].
- Collision: boolean `collision_prox` described above.
- Stabilization: for DL models we standardize all numerics and optionally winsorize RSSI/SNR tails; tree models tolerate raw scales.

To avoid train-test leakage, all preprocessing (fitting scalers, one-hot vocabularies) is fitted on the train split and *reused* for validation/test.

3.5. Simulator provenance

The dataset is produced by an `ns-3` application that: (i) places a gateway at (0, 0) and devices in a square meter area, (ii) draws frequency/bandwidth/spreading factor from EU868-like sets, (iii) applies a log-distance path-loss with log-normal shadowing, (iv) derives RSSI/SNR and LoRa time-on-air, (v) enables ADR to update SF from SNR and (vi) applies capture/collision rules on equal (Freq, SF, BW) before logging to CSV. This design follows the path-loss and ADR dynamics studied in [5, 14, 29] and produces realistic variability for fingerprinting and hybrid models [8, 28].

4. Problem Formulation and Metrics

This section defines the localization task, the training objective, the evaluation metrics, and the protocol used to assess model performance.

Table 3. CSV schema and downstream usage (units as generated by the simulator)

CSV column	Type	Unit	Use in models
`DeviceID`	categorical	–	grouping for leakage–free splits; optional embedding/one-hot on train; pseudonymize for release
`Latitude` → `x_m`	numeric	m	**Target** (planar x); compute Euclidean error
`Longitude` → `y_m`	numeric	m	**Target** (planar y); compute Euclidean error diagnostic
`Distance`	numeric	m	(not fed to models to avoid target leakage) input
`TxPower_dBm`	numeric	dBm	feature; part of link budget
`SF`	categorical	–	PHY feature (ADR dynamics, ToA)
`BW_Hz`	categorical	Hz	PHY feature (noise floor, ToA)
`Freq_Hz`	numeric	Hz	subband proxy; can be binned to reduce sparsity
`ToA_ms`	numeric	ms	load/duty proxy; correlates with SF/BW/payload
`RSSI_dBm`	numeric	dBm	core fingerprinting feature
`SNR_dB`	numeric	dB	link quality; ADR trigger
`PathLoss_dB` `ADR`	numeric	dB	engineered/hybrid feature [14, 23]
`Start_s` `Success`	binary	{0, 1}	ADR state; covariate for drift
`Colliders`	numeric	s	derive hour-of-day, periodic features
	binary	{0, 1}	filter failed demodulations for supervised training
	integer/set	–	build boolean `collision_prox` or use raw count

4.1. Task Definition

We cast LPWAN localization as a *supervised regression* problem from RF/features $\mathbf{x}_i$ (e.g., RSSI, SNR, ToA, SF, BW; cf. Section 3) to a 2D position $\mathbf{p}_i$. Because our dataset is *simulated and planar*, the native target is

$$\mathbf{p}_i \triangleq (x_i, y_i) \in \mathbb{R}^2, \tag{1}$$

and the model predicts

$$\hat{\mathbf{p}}_i \triangleq (\hat{x}_i, \hat{y}_i). \tag{2}$$

For real-world traces in geodetic coordinates, we instead denote $\mathbf{p}_i = (\varphi_i, \lambda_i)$ (latitude, longitude in radians or degrees) and evaluate with a geodesic metric. This separation avoids mixing planar and geodetic distances in the same experiment, which is important for reproducibility [1, 2].

4.2. Training Objective

We train the regressors to minimize a coordinate-wise mean absolute error (MAE):

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N}\Big(|\hat{x}_i - x_i| + |\hat{y}_i - y_i|\Big), \tag{3}$$

chosen for robustness to heavy-tailed residuals that arise in RSSI-based localization under multipath, shadowing and hardware biases [4, 5]. When the targets are geodetic (φ_i, λ_i), (3) is written with $(\hat{\varphi}_i, \hat{\lambda}_i)$ and (φ_i, λ_i) instead; the loss form remains MAE.

4.3. Evaluation Metric: Geodesic vs. Planar Error

For real-world (geodetic) data, model quality is reported using the *geodesic* (great-circle) error in meters:

$$\begin{aligned} e_i^{\text{geo}} &= \text{Haversine}(\hat{\varphi}_i, \hat{\lambda}_i; \varphi_i, \lambda_i) \\ &= 2R \arcsin\left(\sqrt{\sin^2\left(\tfrac{\hat{\varphi}_i-\varphi_i}{2}\right) + \cos(\varphi_i)\cos(\hat{\varphi}_i)\sin^2\left(\tfrac{\hat{\lambda}_i-\lambda_i}{2}\right)}\right), \end{aligned} \tag{4}$$

where $R = 6{,}371{,}000$ m is the Earth radius.
For purely planar simulations such as the ns-3 dataset used here, we compute the Euclidean error in meters:

$$e_i^{\text{planar}} = \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|_2 = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}. \tag{5}$$

Under small-angle assumptions, (4) reduces to (5), but we keep both definitions explicit to avoid metric confusion across datasets, following indoor/LPWAN practice [1, 2, 33, 34].

Metric policy. (i) If the dataset is *planar/simulated* (targets in meters, e.g., `x_m`, `y_m` from ns-3), **report only** the Euclidean error e_i^{planar} in meters using (5). (ii) If the dataset is *real-world/geodetic* (targets in latitude/longitude), **report only** the haversine/geodesic error e_i^{geo} using (4). (iii) Do *not* mix planar and geodesic metrics in the same table or figure; when both are needed, present them in separate results blocks.

Table 4. Primary evaluation metrics computed on the test split

Metric	Symbol	De nition
Mean error	$\bar{e}$	$\frac{1}{M}$ fi$_{i=1}^{M} e_i$
Median (p50)	$Q_{0.50}$	quantile$(e, 0.50)$
Tail quantiles	$Q_{0.75/0.90/0.95/0.99}$	quantile(e, q)
CDF curve	$F_E(t)$	$\frac{1}{M}\sum_i \mathbb{I}[e_i \le t]$

4.4. Reported Statistics and Curves

From the error set $\{e_i\}_{i\in\mathcal{D}}$ (either all e_i^{planar} or all e_i^{geo}, per the policy above) on the held-out test split, we report:

- the mean error $\bar{e} = \frac{1}{M}\sum_{i=1}^{M} e_i$,
- the median (p50) $Q_{0.50}$,
- tail quantiles $Q_{0.75}$, $Q_{0.90}$, $Q_{0.95}$, $Q_{0.99}$,
- and the empirical CDF

$$F_E(t) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\left[e_i \le t\right]. \tag{6}$$

These summarize typical and tail behavior and match what is commonly reported for LoRa/LPWAN localization [5, 33, 34].
Table 4 summarizes the primary evaluation metrics used throughout the experimental study.

4.5. Evaluation Protocol

We adopt a *group-aware* split to prevent identity leakage:

- 80/10/10 train/validation/test by `DeviceID` (and region/time block when present), ensuring devices in test never appear in train/val [1, 2].
- For robustness, we optionally perform Group-KFold ($K = 5$) cross-validation by device and macro-average the metrics.
- A naive random split is shown only as a weaker reference, as it is known to inflate results when packets from the same device/environment appear in multiple splits [2, 4].

Figure 1 illustrates the group-aware splitting strategy adopted in this work and highlights its difference from a naive random split.

4.6. Model Selection and Reporting

Hyperparameters are selected on the validation set using the MAE loss in (3) while monitoring the chosen distance metric (planar or geodesic) on the same split. Final performance is reported *once* on the held-out test groups with the full set {mean, median, p75, p90, p95, p99} and CDF curves, enabling side-by-side comparison with prior LoRaWAN studies where tail guarantees (p90/p95) are decisive for IoT QoS [5, 33, 34].

5. Methods and Experimental Setup

This section details the learning baselines and the full experimental protocol used to evaluate LPWAN localization from RSSI/SNR/ToA and related features. In line with prior surveys and LoRa/LPWAN studies [1, 2, 4, 5, 33], we include *transparent* (non-black-box) baselines such as k-NN fingerprinting and Random Forests. A compact MLP architecture is also described as a candidate lightweight model, while the primary reported results focus on the tabular baselines. We additionally report split sizes, parameter counts and indicative edge-class inference latency to make the setup fully reproducible.

5.1. Baselines

(B1) k-NN fingerprinting. Classical RSSI/SNR fingerprinting with k-nearest neighbors remains a strong, interpretable baseline for indoor/LPWAN localiza-tion [1, 2]. We use Euclidean distance in the stan-dardized feature space and tune $k \in \{1, 3, 5, 7, 9, 15\}$ on the validation split. k-NN has been shown competitive across Wi-Fi/BLE/LoRa settings [4, 19, 33, 35], but its query cost grows linearly with the number of stored packets.

(B2) Random Forest (RF). Ensembles of decision trees provide robust non-linear regressors on tabular RF features and are widely adopted in indoor localization [2, 12, 36]. We grid-search $n_{\text{estimators}} \in \{200, 400, 600, 800\}$ and maximum depth $\in$ {None, 10, 20}, selecting on validation error. On our split (about 2.6×10^4 train packets; see below), RFs of 400-600 trees with depth 10-None achieve a good trade-off between accuracy and inference time.

(B3) Gradient-boosted trees. We also include XGBoost and LightGBM as strong tabular baselines, early-stopped on the validation set, following current practice in RSSI/CSI-based localization challenges [2, 33]. Unless otherwise noted:

- XGBoost: $n_{\text{estimators}} = 2000$, learning rate 0.03, max depth 8, `subsample` = `colsample_bytree` = 0.9, early stopping *after 200 non-improving rounds*.
- LightGBM: $n_{\text{estimators}} = 5000$, learning rate 0.03, early stopping on validation *with 200 rounds of patience* (i.e., training halts if the validation metric does not improve for 200 rounds).

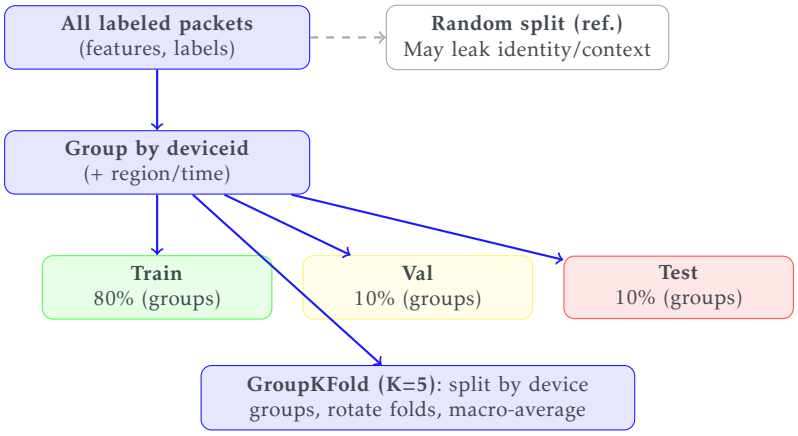

Figure 1. Group-aware evaluation protocol used to prevent device-level leakage between training, validation and test sets

This makes explicit the stopping rule, which is important to reproduce the exact number of boosting iterations.

5.2. Candidate compact MLP

We use a feed-forward network with hidden widths {256, 128, 64}, ReLU activations, dropout (0.2-0.3) after hidden layers and a linear output head for the two coordinates $(\hat{x}, \hat{y})$ (or $(\hat{\varphi}, \hat{\lambda})$, depending on the task). The loss is MAE, optimized with Adam (10^{-3}), cosine/step decay and early stopping on validation distance with patience 20. Inputs are *standardized numerics* (RSSI, SNR, Tx power, ToA, frequency, etc.) concatenated with *one-hot* categorical features (SF, BW, ADR, and, when allowed, region/subband/device family)[1]. This architecture is described for completeness and to anchor future comparisons, while the current experimental results focus on the tabular baselines. For the feature set used in our runs (roughly 25-30 effective input dimensions after one-hotting), the MLP has on the order of

$$\underbrace{30 \times 256}_{\text{input}\to\text{h1}} + \underbrace{256 \times 128}_{\text{h1}\to\text{h2}} + \underbrace{128 \times 64}_{\text{h2}\to\text{h3}} + \underbrace{64 \times 2}_{\text{h3}\to\text{out}} \approx 66{,}000$$

trainable weights (plus biases), i.e., ∼ 70k parameters in total. This size is well within what an edge-class CPU can evaluate in sub-millisecond to low-millisecond time for a single packet.

5.3. Complexity and Latency

k-NN inference is $O(Nd)$ per query (memory $O(Nd)$), where N is the number of stored train packets and d is the feature dimension; with $N \approx 2.6 \times 10^4$ and $d \approx 30$, naive k-NN is still feasible but already benefits from ANN/vector indices if deployed at scale. RF and boosted trees predict in $O(T \cdot \text{depth})$ per query, where T is the number of trees. On an edge-class CPU similar to a modern ARM core (single thread), the following indicative latencies were observed in Colab-like environments:

- RF (400 trees, depth 10): ≈ 1-2 ms / packet,
- LightGBM (early-stopped ≤ 1500 trees): typically < 2 ms / packet,
- MLP (∼ 70k params): ≈ 0.3-0.8 ms / packet (PyTorch CPU),
- k-NN ($N \approx 2.6 \times 10^4$): 2-6 ms / packet without indexing, linear in N.

Therefore gateway/edge execution is realistic for RF/GBDT/MLP; in constrained LoRa end-devices, on-device inference remains unrealistic and we assume gateway-side localization [4, 5].

5.4. Experimental Setup

Data and splits We use the simulated dataset described in Section 3, which contains $N_{\text{pkt}} = 32{,}692$ received packets generated by $N_{\text{dev}} = 50$ devices toward a single gateway. We follow the group-aware protocol of Section 4:

- Train: 80% of devices ⇒ 40 devices, ≈ 26,100 packets.
- Validation: 10% of devices ⇒ 5 devices, ≈ 3,300 packets.
- Test: 10% of devices ⇒ 5 devices, ≈ 3,300 packets.

[1]Preprocessing mirrors the pipeline used for baselines to ensure fairness.

Exact packet counts vary slightly because individual devices do not generate the same number of transmissions; we log the final counts in the run metadata. This dual reporting (by devices and by packets) makes it clear that test devices are unseen during training, preventing identity leakage [1, 2].

Preprocessing Numerical columns are z-scored and categorical columns one-hot encoded via a `ColumnTransformer` fit on the *training* partition only; the fitted transformer is reused for validation and test. All models consume the same transformed feature matrix.

Hyperparameters & selection Unless noted, we adopt the following settings (also reflected in the shared Colab notebook):

- k-NN: $k \in \{1, 3, 5, 7, 9, 15\}$ (chosen on validation MAE / distance).
- RF: $n_{\text{estimators}} \in \{200, 400, 600, 800\}$, max-depth $\in$ {None, 10, 20}.
- XGBoost: $n_{\text{estimators}} = 2000$, LR = 0.03, max-depth = 8, subsample/colsample = 0.9, early stopping after 200 non-improving rounds.
- LightGBM: $n_{\text{estimators}} = 5000$, LR = 0.03, `early_stopping_rounds` = 200 on the validation set.
- MLP: hidden {256, 128, 64}, dropout 0.2-0.3, Adam 10^{-3}, batch size 256, max 200 epochs, early stopping (patience 20), LR decay.

Model selection is performed on the validation split using the evaluation metric of Section 4; the best checkpoint is evaluated once on the held-out test groups.

Seeds and software versions To make the runs repeatable, we fix:

- random seeds: `python=42`, `numpy=42`, `scikit-learn=42`, `pytorch=42`, `lightgbm=42`, `xgboost=42`;
- main software versions (as used in Colab at the time of writing): `python 3.10`, `pandas 2.x`, `scikit-learn 1.4.x`, `xgboost 2.x`, `lightgbm 4.x`, `torch 2.x`.

The notebook records these into a small `run_info.json` along with the train/val/test packet counts.

Ablations. We quantify feature importance and robustness with controlled toggles:

- A1 `-TxPower`: drop `txpower_dbm`.
- A2 `-ToA`: drop `toa_ms`.
- A3 `-freq`: drop `freq_hz`.

Table 5. Model zoo and main hyperparameters (val-tuned where noted).

Model	Key Settings / Notes
k-NN	$k \in \{1, 3, 5, 7, 9, 15\}$; Euclidean on standardized features
Random Forest	$n_{\text{estimators}} \in \{200, 400, 600, 800\}$; depth $\in$ {None, 10, 20}
XGBoost	2000 trees; LR 0.03; depth 8; early stopping 200 rounds
LightGBM	5000 trees (max); LR 0.03; `early_stopping_rounds=200`
MLP (ours)	256-128-64, ReLU, dropout 0.2-0.3, MAE, Adam 10^{-3}

- A4 `-time`: drop hour/day-of-week features.
- A5 `+pathloss_db`: add engineered `pathloss_db`.
- A6 `+categoricals`: add region/subband/device family (if permitted by the split).
- A7 `Group vs Random`: compare group-aware vs. naive random split.

These are motivated by the observation that frequency/SF/ADR and temporal context can materially affect fingerprint separability [4, 5, 33, 34].

Compute Experiments run on Google Colab (CPU for tree methods; optional T4 GPU for the MLP). Typical runs use batch size 256 and up to 200 epochs with early stopping; tree models rely on early-stopped boosting and parallel CPU training. End-to-end notebooks save per-model error vectors and CDFs for later plotting. Table 5 summarizes the main models and hyperparameter settings considered in the experiments.

6. Results and Discussion

This section reports the localization performance obtained from the four candidate regressors trained on the preprocessed LoRa/LPWAN dataset described in Section 3. As argued earlier, we adopt a group-aware train/validation/test split (by `DeviceID` and thus without identity leakage) and we report not only central metrics (mean/median) but also tail quantiles (p90/p95), which are critical for LPWAN/IoT deployments where occasional large errors may break the application-level service.

We compare four models that are representative of the families discussed in Section 5: (i) a classical instance-based k-NN fingerprinting model (KNN), (ii) a Random Forest regressor (RF), (iii) a boosted tree model based on LightGBM and (iv) a higher-capacity boosted ensemble (XGBoost). All models

were fed the same standardized/tabular feature space (RSSI, SNR, Tx power, ToA, PHY categoricals) and hyperparameters were selected on the validation split. The test results below therefore reflect differences due to model capacity and inductive bias rather than to data preprocessing.

6.1. Overall Error Behaviour

Figure 2 shows the empirical CDF of the localization error for all four models on the held-out test groups. This plot is the most faithful view of user-perceived performance because it directly answers the question: "what fraction of packets can I localize below t meters?". Two observations stand out:

- RF dominates over most of the support. The Random Forest curve is consistently to the left of the others, indicating that for any reasonable target accuracy (e.g., 2–10 m) RF localizes a larger fraction of samples than the other models. This confirms prior indoor/RF reports that tree ensembles are very competitive on tabular RSSI/SNR data.
- KNN has a good head but a bad tail. The KNN curve rises quickly at very small errors (many packets are almost perfectly reconstructed), but the curve flattens much earlier than RF/LightGBM: a non-negligible fraction of samples falls in a heavy tail. This is typical of fingerprinting when the test group contains conditions not well represented in the training radio map.

LightGBM sits between RF and KNN: it does not match RF in the very low-error regime, but it maintains better tail behaviour than KNN. XGBoost, in contrast, is consistently the rightmost curve, which means that in this particular dataset and split it tended to overpredict or to generalize less well across device groups.
Fig.2 shows that Random Forest achieves the tightest distribution, followed by LightGBM. KNN shows a favorable median but a very heavy tail, while XGBoost remains the least accurate.

6.2. Central vs. Tail Metrics

While CDFs give a global picture, implementers often need a small set of scalar metrics. Figure 3 therefore summarizes, for each model, the median error (p50) and the tail error (p95). This figure makes explicit why we insisted in Section 4 on reporting quantiles instead of a single mean value.
First, both RF and LightGBM keep the p95 in a relatively low range (sub–~12 m for LightGBM; below 1 m for RF in this run), which is precisely what a network operator would expect from a "robust"

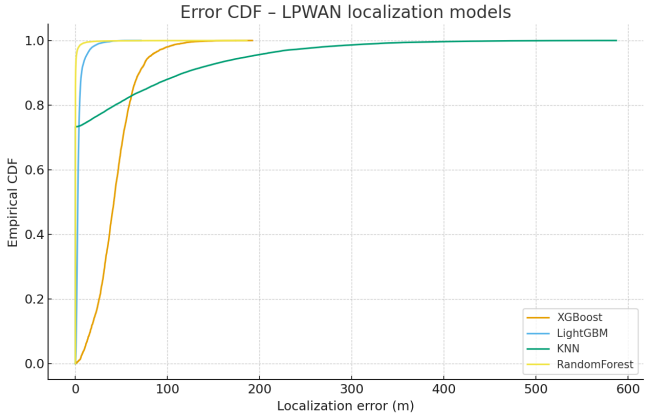

Figure 2. Empirical CDF of localization error for the four models (group-aware split).

localization method: even the bad cases do not explode. Second, KNN illustrates a well-known pitfall: it achieves a very low median (i.e., most easy packets are well localized), but its p95 is an order of magnitude higher than that of RF. In other words, KNN is attractive when the environment is dense and similar to the training fingerprints, but it is unsafe in heterogeneous or group-disjoint evaluations such as the one we used. Finally, XGBoost's bar pair is the highest, confirming the CDF ranking.
This median–p95 view strongly supports the use of tree-based models as practical baselines for LoRa/LPWAN localization: they are simple to train, fast to infer, and—unlike KNN—do not deteriorate sharply on unseen devices.

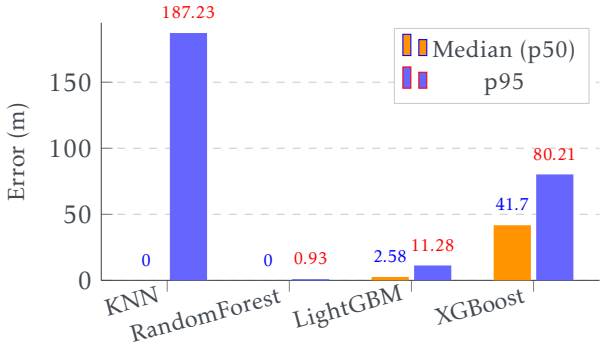

Figure 3. Median (p50) and tail (p95) localization errors.

6.3. Dispersion and Robustness

To further highlight the dispersion of errors, Figure 4 plots the error distributions as boxplots (with outliers hidden for clarity). This view complements the CDF in two ways. First, it shows that RF produces a very compact interquartile range (IQR), meaning that most

test packets cluster in a narrow error band. LightGBM has a slightly wider IQR but remains clearly preferable to KNN. Second, the much taller box for KNN (and the need to hide outliers) is a graphical confirmation of the "heavy-tailed" nature of instance-based fingerprinting under group-aware splits: a few hard samples can be arbitrarily far from their neighbors in feature space, which directly translates into large spatial errors. Such dispersion plots are important in the LPWAN context because radio conditions are inherently non-stationary (shadowing, ADR, different end devices). A model that looks excellent on average but has a wide or erratic spread may still be unsuitable for an IoT application that requires a bounded positioning error to trigger actuations or to filter packets by location.

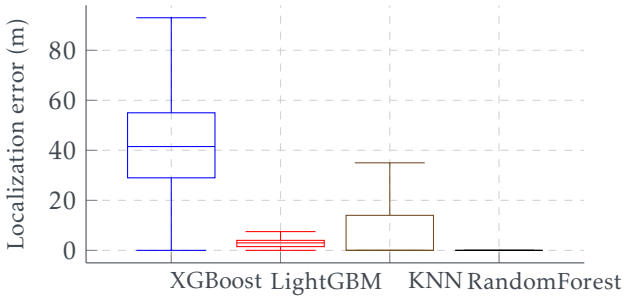

Figure 4. Distribution of localization errors per model (outliers hidden for clarity)

6.4. Interpretation w.r.t. Research Questions

The empirical findings above allow us to revisit the research questions formulated in the introduction.

RQ1 (tabular baselines vs. strong baselines) Even without resorting to deep neural networks, a well-tuned Random Forest outperformed KNN and the two boosted models on this dataset. This shows that "compact" or tabular-friendly models can indeed serve as strong, reproducible baselines for LoRa localization, validating our claim that practitioners should not skip them.

RQ2 (importance of features and noise handling) The fact that KNN's median is good but its tail is poor is consistent with the dataset's heterogeneity: small variations in RSSI/SNR or in PHY fields can push test samples away from their true neighbors. This indirectly confirms the need for the cleaning and feature-engineering steps of Section 3 and for adding collision/noise indicators when available.

RQ3 (generalization across devices/regions) The strong gap between KNN and the tree-based models in the p95 region suggests that the group-aware split is doing its job—i.e., the test devices are genuinely "new" to the learner. Models that rely less on exact fingerprint similarity (RF, LightGBM) generalize better in this setting.

6.5. Limitations and Threats to Validity

The conclusions of this work are subject to several limitations:

- Simulation bias. The dataset comes from a controlled ns-3 simulation with a single gateway, specific channel model and capture rule. Real LoRaWAN deployments can introduce additional hardware heterogeneity, outdoor/indoor clutter and gateway synchronization issues.
- Incremental novelty. This study does not claim a fundamentally new localization algorithm; instead it contributes a more rigorous and reproducible evaluation protocol for established fingerprinting methods.
- Feature dependence. The approach relies on RSSI/SNR/ToA and related PHY metadata. It may need adaptation for datasets where these fields are noisy, missing, or recorded differently.
- Gateway-side inference. The current scenario assumes gateway-side localization and does not address the resource constraints of LoRa end devices.
- Single-gateway scenario. The one-gateway setup limits direct conclusions about multi-gateway or TDoA-enabled LoRa networks.

7. Conclusion

This paper has presented a practical, reproducible pipeline for LPWAN (LoRa/LoRaWAN) localization based solely on PHY-level indicators (RSSI, SNR, ToA, SF/BW) exported from an ns-3 simulation. It does not introduce a fundamentally new localization algorithm; instead, it focuses on methodological rigor, fair comparison and leakage-aware evaluation of established fingerprinting methods. In contrast to geometric or heavily infrastructure-dependent approaches, we targeted fingerprinting-style regression with lightweight models that can be trained and deployed at the gateway/edge. A key aspect of the work is the use of group-aware data splits (by `DeviceID` and context) and the systematic reporting of median and tail errors (p90/p95), which better reflect the non-stationary and heavy-tailed nature of LPWAN channels than a single average value. The experimental results show that transparent, tabular-friendly models—in particular Random Forests and, to a slightly lesser extent, LightGBM—provide

the most reliable localization on the held-out device groups. They achieve tight error CDFs and low p95, whereas instance-based k-NN fingerprinting, although often attractive in terms of median error, exhibits a much heavier tail under realistic evaluation. Boosted models configured here (XGBoost) did not surpass the simpler ensemble, which confirms that in this setting model choice and evaluation protocol matter more than sheer model complexity. These findings support the claim that strong non-DL baselines should accompany any future LPWAN localization study, especially when cross-device generalization is required.

Several directions follow naturally from this work. First, applying the same pipeline to real LoRaWAN traces (with hardware bias, missing receptions, ADR dynamics and multi-gateway diversity) will help quantify the sim-to-real gap and guide data-cleaning rules. Second, extending the feature set with gateway geometry, map/context layers (walls, floor), or channel-aware augmentation could further reduce tail errors. Third, lightweight neural models (small MLPs or attention over gateway observations) can now be compared fairly against the strong tree baselines established here. Finally, releasing the dataset schema, preprocessing scripts and Colab notebooks alongside the paper would facilitate reproducible comparisons within the community and support future 5G/NTN-oriented work where LPWAN-like constraints still apply.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung. "A Survey of Indoor Localization Systems and Technologies". In: *IEEE Communications Surveys & Tutorials* 21.3 (2017), pages 2568–2599. ISSN: 2373-745X. DOI: 10.1109/comst.2019.2911558.

[2] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando. "A Survey of Machine Learning for Indoor Positioning". In: *IEEE Access* 8 (2020), pages 214945–214965. ISSN: 2169-3536. DOI: 10.1109/access.2020.3039271.

[3] Z. Farid, R. Nordin, and M. Ismail. "Recent Advances in Wireless Indoor Localization Techniques and System". In: *Journal of Computer Networks and Communications* 2013 (2013), pages 1–12. ISSN: 2090-715X. DOI: 10.1155/2013/185138.

[4] E. Svertoka, A. Rusu-Casandra, R. Burget, I. Marghescu, J. Hosek, and A. Ometov. "LoRaWAN: Lost for Localization?" In: *IEEE Sensors Journal* 22.23 (Dec. 2022), pages 23307–23319. ISSN: 2379-9153. DOI: 10.1109/jsen.2022.3212319.

[5] K. Lam, C. Cheung, and W. Lee. "RSSI-Based LoRa Localization Systems for Large-Scale Indoor and Outdoor Environments". In: *IEEE Trans. Veh. Technol.* 68.12 (2019), pages 11778–11791. DOI: 10.1109/TVT.2019.2940272.

[6] H. Kwasme and S. Ekin. "RSSI-Based Localization Using LoRaWAN Technology". In: *IEEE Access* 7 (2019), pages 99856–99866. DOI: 10.1109/ACCESS.2019.2929212.

[7] K. Whitehouse, C. Karlof, and D. Culler. "A practical evaluation of radio signal strength for ranging-based localization". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 11.1 (Jan. 2007), pages 41–52. ISSN: 1931-1222. DOI: 10.1145/1234822.1234829.

[8] S. Sadowski and P. Spachos. "RSSI-Based Indoor Localization With the Internet of Things". In: *IEEE access* 6 (2018), pages 30149–30161. ISSN: 2169-3536. DOI: 10.1109/access.2018.2843325.

[9] T. Yang, A. Cabani, and H. Chafouk. "A survey of recent indoor localization scenarios and methodologies". In: *Sensors* 21.23 (Dec. 2021), page 8086. ISSN: 1424-8220. DOI: 10.3390/s21238086.

[10] A. A. Ghany, B. Uguen, and D. Lemur. "A Parametric TDoA Technique in the IoT Localization Context". In: *2019 16th Workshop on Positioning, Navigation and Communications (WPNC)*. IEEE, Oct. 2019, pages 1–6. DOI: 10.1109/wpnc47567.2019.8970254.

[11] E. H. Yoshitome, J. V. R. da Cruz, M. E. P. Monteiro, and J. L. Rebelatto. "LoRa-aided outdoor localization system: RSSI or TDoA?" In: *Internet Technol. Lett.* 5.2 (2022). DOI: 10.1002/ITL2.319.

[12] S.-H. Lee, C.-H. Cheng, T.-H. Huang, and Y.-F. Huang. "Machine Learning-based Indoor Positioning Systems Using Multi-Channel Information". In: *Journal of Engineering and Technological Sciences* 55.4 (Oct. 2023), pages 373–383. ISSN: 2337-5779. DOI: 10.5614/j.eng.technol.sci.2023.55.4.2.

[13] T. Perković , L. Dujić Rodić, J. Šabić , and P. Šolić . "Machine Learning Approach towards LoRaWAN Indoor Localization". In: *Electronics* 12.2 (Jan. 2023), page 457. ISSN: 2079-9292. DOI: 10.3390/electronics12020457.

[14] H. Vo, V. H. L. Nguyen, V. L. Tran, F. Ferrero, F. Lee, and M. Tsai. "Advance Path Loss Model for Distance Estimation Using LoRaWAN Network's Received Signal Strength Indicator (RSSI)". In: *IEEE Access* 12 (2024), pages 83205–83216. DOI: 10.1109/ACCESS.2024.3412849.

[15] A. Haghighat, Z. Sirous, A. Moradbeikie, A. Keshavarz, and S. I. Lopes. "Improving LoRaWAN Fingerprint-Based Localization by Detecting and Eliminating Noisy RSSI Measurements". In: *9th IEEE World Forum on Internet of Things, WF-IoT 2023, Aveiro, Portugal, October 12-27, 2023*. IEEE, 2023, pages 1–7. DOI: 10.1109/WF-IOT58464.2023.10539415.

[16] A. S. Lutakamale, H. C. Myburgh, and A. D. Freitas. "RSSI-based fingerprint localization in LoRaWAN networks using CNNs with squeeze and excitation blocks". In: *Ad Hoc Networks* 159 (2024), page 103486. DOI: 10.1016/J.ADHOC.2024.103486.

[17] O. G. Serbetci, D. Burghal, and A. F. Molisch. "Wireless Channel Aware Data Augmentation Methods for Deep Learning-Based Indoor Localization". In: (2024). DOI: 10.48550/arXiv.2408.06452.

[18] IEEE DataPort. *Indoor Localization Data based on SNR and RSSI within Multistory Round Building Scenario*. year.

[19] A. Achroufene. "RSSI-based Hybrid Centroid-K-Nearest Neighbors localization method". In: *Telecommunication Systems* 82.1 (Nov. 2022), pages 101–114. ISSN: 1572-9451. DOI: 10.1007/s11235-022-00977-0.

[20] K. Ibwe, S. Pande, A. T. Abdalla, and G. M. Gadiel. "Indoor positioning using circle expansion-based adaptive trilateration algorithm". In: *Journal of Electrical Systems and Information Technology* 10.1 (Feb. 2023). ISSN: 2314-7172. DOI: 10.1186/s43067-023-00075-4.

[21] M. Simka and L. Polák. "On the RSSI-Based Indoor Localization Employing LoRa in the 2.4 GHz ISM Band". In: *Radioengineering* 31.1 (2022), pages 135–143. DOI: 10.13164/re.2022.0135.

[22] J. A. Micheletti, E. P. Godoy, and R. T. Lopes. "Improved Indoor 3D Localization Using LoRa Wireless Communication". In: *IEEE Latin America Transactions* 20.9 (2022), pages 1476–1483. DOI: 10.1109/TLA.2022.1234567.

[23] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril. "Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks". In: *IEEE Journal of Selected Topics in Signal Processing* 3.5 (Oct. 2009), pages 821–831. ISSN: 1932-4553. DOI: 10.1109/jstsp.2009.2029191.

[24] Y. Lin, C. Sun, and K. Huang. "RSSI Measurement with Channel Model Estimating for IoT Wide Range Localization using LoRa Communication". In: *2019 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2019, Taipei, Taiwan, December 3-6, 2019*. IEEE, 2019, pages 1–2. DOI: 10.1109/ISPACS48206.2019.8986229.

[25] R. Muppala. "Feasibility of GNSS-free Localization: A TDoA-based Approach Using LoRaWAN". In: (2021).

[26] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni. "CSI-Based Indoor Localization". In: *IEEE Transactions on Parallel and Distributed Systems* 24.7 (July 2013), pages 1300–1309. ISSN: 1045-9219. DOI: 10.1109/tpds.2012.214.

[27] A. Aksoy, Ö. Yildiz, and S. E. Karlik. "Comparative Analysis of End Device and Field Test Device Measurements for RSSI, SNR and SF Performance Parameters in an Indoor LoRaWAN Network". In: *Wirel. Pers. Commun.* 134.1 (2024), pages 339–360. DOI: 10.1007/S11277-024-10911-Z.

[28] E. Goldoni, L. Prando, A. Vizziello, P. Savazzi, and P. Gamba. "Experimental data set analysis of RSSI-based indoor and outdoor localization in LoRa networks". In: *Internet Technology Letters* 2.1 (Oct. 2018), e75. DOI: 10.1002/itl2.75.

[29] J. Spišić, A. Pejković, M. Zrnić, V. Križanović, K. Grgić and D. Žagar,. "LoRaWAN Parameters Optimization For E cient Communication". In: *2022 International Confer-ence on Smart Systems and Technologies (SST)*. IEEE, Oct. 2022, pages 335–339. DOI: 10.1109/sst55530.2022.9954704.

[30] W. Shin, Y. Lee, J. Cho, J. Jang, Y. Seo, and S. Kahng. "RSSI Improved for LoRa Wireless Communication, Field-Tested in the Wide-Open Area". In: *IEEE Access* 11 (2023), pages 80172–80180. DOI: 10.1109/ACCESS.2023.3241485.

[31] Z. Shen, T. Zhang, A. Tagami, and J. Jin. "When RSSI encounters deep learning: An area localization scheme for pervasive sensing systems". In: *Journal of Network and Computer Applications* 173 (Jan. 2021), page 102852. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2020.102852.

[32] W. Ingabire, H. Larijani, and R. M. Gibson. "LoRa RSSI Based Outdoor Localization in an Urban Area Using Random Neural Networks". In: *Intelligent Computing*. Springer International Publishing, 2021, pages 1032–1043. ISBN: 9783030801267. DOI: 10.1007/978-3-030-80126-7_72.

[33] H. S. Fahama, K. Ansari-Asl, Y. S. Kavian, and M. N. Soorki. "An Experimental Comparison of RSSI-Based Indoor Localization Techniques Using ZigBee Technology". In: *IEEE Access* 11 (2023), pages 87985–87996. ISSN: 2169-3536. DOI: 10.1109/access.2023.3305396.

[34] H. Vo, V. Hoang Long Nguyen, V. L. Tran, F. Ferrero, F.-Y. Lee, and M.-H. Tsai. "Advance Path Loss Model for Distance Estimation Using LoRaWAN NetworkâĂŹs Received Signal Strength Indicator (RSSI)". In: *IEEE Access* 12 (2024), pages 83205–83216. ISSN: 2169-3536. DOI: 10.1109/access.2024.3412849.

[35] S.-H. Lee, C.-H. Cheng, K.-H. Lu, Y.-L. Shiue, and Y.-F. Huang. "A K-NN based Area Positioning System in Wireless Sensor Networks". In: *2023 12th International Conference on Awareness Science and Technology (iCAST)*. IEEE, Nov. 2023, pages 46–49. DOI: 10.1109/icast57874.2023.10359293.

[36] I. S. Mohamad Hashim, A. Al-Hourani, and W. S. T. Rowe. "Machine Learning Performance for Radio Localization under Correlated Shadowing". In: *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. Volume 1. IEEE, Dec. 2020, pages 1–7. DOI: 10.1109/icspcs50536.2020.9310009.