

A Distributed Range-Free Localization WSN Deployment Using Machine Learning Techniques for Air Pollution Monitoring

Anil Kumar K^{1*}, Prabhakar M²

¹ Research Scholar, School of CSE, REVA University, Bengaluru, India

² Professor, School of CSE, REVA University, Bengaluru, India

HOW TO CITE:

Anil Kumar K, Prabhakar M,
(2026). A Distributed Range-Free
Localization WSN Deployment
Using Machine Learning
Techniques for Air Pollution
Monitoring.
International Journal of Special
Education, 41(12s), 1240-1260

ABSTRACT:

Accurate and energy-efficient localization of wireless sensor nodes is crucial for reliable air quality monitoring in cities. Traditional range-based methods such as Time of Arrival (ToA) and Received Signal Strength Indicator (RSSI) are limited by hardware complexity and the need for calibration, making them less suitable for large-scale deployments. While range-free algorithms are computationally simpler, they are prone to cumulative hop-count errors and irregular radio signals, which reduce accuracy. This paper presents a new framework called Distributed Machine Learning Assisted Range-Free Localization (DML-RFL). It merges anchor-assisted distributed localization with ensemble machine learning techniques to achieve highly precise node positioning in urban settings. DML-RFL begins with DV-Hop and Centroid Localization for initial estimates, then refines these by inputting features into models like Random Forest, Support Vector Machine, Gradient Boosting, and Artificial Neural Networks. The framework is structured within a five-layer architecture that includes sensor, communication, edge processing, machine learning inference, and environmental monitoring dashboard layers, each tailored to support diverse pollution sensors such as PM_{2.5}, PM₁₀, CO, CO₂, SO₂, NO₂, and O₃. Extensive simulations across network densities of 100 to 500 nodes and deployment areas from 500×500 m² to 1500×500 m² show that DML-RFL attains a mean localisation error of 0.31 m in optimal setups, extends network lifetime by 37.4% compared to DV-Hop, and achieves over 97.2% packet delivery ratio. Comparative tests against DV-Hop, APIT, Centroid Localisation, and Weighted Centroid Localisation confirm statistically significant improvements in all key metrics. The proposed system is scalable, cost-effective, and ready for deployment in next-generation urban air-pollution monitoring systems.

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the
terms and conditions
of the Creative Commons Attribution
(CC BY)
license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Wireless Sensor Networks, Range-Free Localization, DV-Hop, Machine Learning, Air Pollution Monitoring, Edge Computing, Internet of Things, DML-RFL, Urban Sensing, Distributed Systems

1. Introduction

The rapid expansion of urban development has rendered air quality monitoring an indispensable component of smart city infrastructure. Concentrations of particulate matter, ground-level ozone, sulphur dioxide, nitrogen dioxide, and carbon monoxide collectively present significant health hazards, with the World Health Organization approximating that approximately seven million premature deaths occur annually due to ambient air pollution [1]. While conventional centralised monitoring stations provide precise data, their elevated installation costs and limited spatial coverage render them inadequate for comprehensive epidemiological analysis and real-time hazard detection [2].

Wireless Sensor Networks (WSNs) have gained widespread recognition as an effective solution for dense, cost-efficient, and energy-conserving environmental sensing. A conventional air quality monitoring WSN typically comprises hundreds or thousands of heterogeneous sensor nodes equipped with chemical and particulate sensors, collaboratively measuring environmental parameters and transmitting data to sink nodes or cloud platforms via multi-hop communication protocols [3]. The precise localization of nodes is crucial within these systems, as pollution mapping, gradient analysis, and source detection algorithms all depend on the accurate geographic coordinates of each sensor [4].

Localisation in Wireless Sensor Networks (WSNs) falls into two main categories: range-based and range-free methods. Range-based approaches, including ToA, TDoA, and RSSI triangulation, deliver high precision but often demand specialised hardware, precise time synchronization, and regular calibration [5]. In contrast, range-free techniques like DV-Hop, APIT, Centroid Localisation, and Monte Carlo Localisation are more straightforward and better suited for battery-powered devices, although they typically have localisation errors between 10-30% of the radio range, which may not be adequate for detailed pollution mapping [6].

Machine learning continuously enhances localization accuracy in WSNs by modelling the

complex, non-linear relationships between network features (such as hop counts, anchor connectivity, and RSSI fingerprints) and physical locations, using labeled training data [7]. However, most current methods depend on centralized architectures that can cause communication bottlenecks, privacy issues, and are not ideal for resource-constrained edge devices [8]. While federated and distributed learning approaches are beginning to address these problems, their use in range-free localization for pollution monitoring remains mostly underexplored [9].

This paper addresses the earlier identified gap by proposing the Distributed Machine Learning Assisted Range-Free Localization (DML-RFL) framework. The main contributions of this research are summarised below:

1. Developed a distributed, range-free localization framework suitable for large urban WSN deployments, eliminating the need for specialized ranging hardware.
2. Incorporated an ensemble machine learning pipeline that includes Random Forest, SVM, Gradient Boosting, and ANN within an edge-distributed inference setup.
3. Created detailed mathematical models for localization error, energy use, communication costs, and network coverage.
4. Designed a five-layer system architecture covering sensing, communication, edge processing, ML inference, and a real-time pollution monitoring dashboard.
5. Conducted extensive simulations across various node densities and area sizes, benchmarking against DV-Hop, APIT, Centroid Localization, and Weighted Centroid Localization.

The remainder of this paper is structured as follows. Section 2 reviews existing research on WSN localization and ML-assisted sensing. Section 3 describes the system model and problem setup. Section 4 introduces the proposed DML-RFL framework. Section 5 details the machine learning components. Section 6 discusses the air pollution monitoring

architecture. Section 7 explains the experimental setup. Sections 8 and 9 present the results and a performance comparison. Section 10 discusses the benefits, limitations, and future directions. Finally, Section 11 provides the conclusion.

2. Literature Review / Related Work

2.1 Range-Free Localization in WSNs

The DV-Hop algorithm, developed by Niculescu and Nath [10], determines node locations by counting hops to anchor nodes and calculating an average hop distance. While it is efficient in terms of computation, DV-Hop may produce large errors in irregular topologies where hop distances vary. He et al. [11] introduced the APIT algorithm, which evaluates whether an unknown node is inside or outside triangular regions created by three anchor nodes, thereby enhancing robustness in sparse networks. Centroid Localization [12], on the other hand, places unknown nodes at the geometric centre of their communicating anchors. This method is easy to implement but tends to be less accurate when anchor nodes are unevenly distributed.

Blumenthal et al. [13] introduced Weighted Centroid Localization, which uses RSSI-based weights for anchor contributions, yielding clear accuracy gains without additional hardware. Monte Carlo Localization techniques [14] employ probabilistic methods to iteratively improve position estimates using mobility models, but their repeated sampling demands computational resources unsuitable for static setups. Range-free localisation methods based on convex optimisation [15] reframe the problem as a semidefinite program, providing bounds on theoretical optimality but requiring substantial processing power.

2.2 Machine Learning for WSN Localization

Since the influential fingerprinting study by Bahl and Padmanabhan [16] on indoor positioning, machine learning's role in WSN localisation has expanded significantly. Liu et al. [17] demonstrated that Random Forest regressors can notably enhance DV-Hop localization accuracy by learning residual correction maps from anchor hop-count data. Payal et al. [18] applied Support

Vector Regression (SVR) for RSSI-based localization, achieving sub-meter accuracy in controlled indoor environments. Gradient Boosting techniques, such as XGBoost, have been used for outdoor WSN localization by Zhao et al. [19], and have shown effectiveness in dealing with non-Gaussian noise and irregular propagation conditions.

Deep learning approaches have gained growing popularity, with Convolutional Neural Networks (CNNs) used for extracting spatial features from radio maps [20], and Long Short-Term Memory (LSTM) networks employed to smooth mobile node trajectories over time [21]. Zhang et al. [22] proposed a federated learning framework for privacy-preserving indoor localization, demonstrating that training models across edge devices causes only minimal accuracy loss compared to centralized approaches. However, the application of these techniques in outdoor pollution monitoring wireless sensor networks (WSNs)—which face constraints in energy and bandwidth—has received less attention.

2.3 Air Quality Monitoring with WSNs

Motlagh et al. [23] outlined a comprehensive UAV-assisted WSN system for urban pollution monitoring, utilising cost-effective electrochemical sensors for NO₂ and O₃ detection and GPS for node localisation. Rao et al. [24] created a LoRaWAN-based WSN to monitor PM_{2.5} over a 15 km² urban area, showcasing the potential for extensive, energy-efficient sensing networks. Yi et al. [25] examined how pollution source localisation algorithms' sensitivity impacts position accuracy and the quality of pollutant maps. Kumar et al. [26] incorporated edge computing into WSN-based air quality monitoring, reducing cloud data transmission by 68% through localised data aggregation and anomaly detection.

2.4 Research Gap and Motivation

Despite the summarized research efforts, a notable gap persists: a framework that integrates distributed range-free localization with machine learning-enhanced coordinate refinement within a dedicated air pollution monitoring WSN

architecture is lacking. Most existing ML-based localization techniques primarily target indoor settings, rely on centralized training, and are evaluated independently of pollution sensing objectives. The proposed DML-RFL framework aims to address these specific challenges.

3. System Model and Problem Formulation

3.1 Network Model

Consider a wireless sensor network (WSN) with N sensor nodes randomly distributed over a two-dimensional area $A = L \times L$, where L is the length of each side. A fraction $\alpha = M/N$ (with M much smaller than N) consists of anchor nodes with precisely known positions, obtained through GPS or manual survey. The remaining $N - M$ nodes are unknown, and their locations must be estimated. All nodes are assumed to operate under an isotropic radio propagation model with a uniform communication radius R_c .

Let $A_i = (x_i, y_i)$ denote the actual position of anchor node i , while $U_j = (x_j, y_j)$ indicates the true location of unknown node j . The estimated position of unknown node j is represented as $\hat{U}_j = (\hat{x}_j, \hat{y}_j)$. Each node is equipped with a multi-species electrochemical sensing module, a radio transceiver, a microcontroller, and an energy storage unit initially holding energy E_0 .

3.2 Communication Model

The first-order radio energy model [27] is adopted. The energy consumed for transmitting k bits over a distance d is given by:

$$E_{tx}(k, d) = k \cdot E_{elec} + k \cdot \epsilon_{amp} \cdot d^2 \quad \dots (1)$$

where $E_{elec} = 50$ nJ/bit is the electronics energy coefficient and $\epsilon_{amp} = 100$ pJ/bit/m² is the amplifier energy coefficient. The energy consumed for receiving k bits is:

$$E_{rx}(k) = k \cdot E_{elec} \quad \dots (2)$$

The total communication energy per sensing round for node j communicating with n_h hop neighbors is:

$$E_{comm}(j) = \sum_{h=1}^{n_h} [E_{tx}(k, d_{j,h}) + E_{rx}(k)] \quad \dots (3)$$

3.3 Localization Error Metrics

The Euclidean distance between the estimated and true coordinates of unknown node j is defined as the localization error ϵ_j :

$$\epsilon_j = \sqrt{(\hat{x}_j - x_j)^2 + (\hat{y}_j - y_j)^2} \quad \dots (4)$$

The normalized localization error (NLE), expressed as a fraction of the communication radius R_c , is:

$$NLE_j = \epsilon_j / R_c \quad \dots (5)$$

The Mean Absolute Error (MAE) across all $N - M$ unknown nodes is:

$$MAE = (1/(N-M)) \cdot \sum_{j=1}^{N-M} |\epsilon_j| \quad \dots (6)$$

The Root Mean Square Error (RMSE) is:

$$RMSE = \sqrt{(1/(N-M)) \cdot \sum_{j=1}^{N-M} \epsilon_j^2} \quad \dots (7)$$

3.4 Network Coverage and Lifetime

The network coverage ratio Ψ is defined as the fraction of the deployment area A that is monitored by at least one active sensor node:

$$\Psi = |A_{covered}| / |A| \quad \dots (8)$$

where $A_{covered}$ is the union of sensing disc areas with radius R_s . The network lifetime Λ is defined as the time until the first node energy depletion:

$$\Lambda = \min_{j \in [1, N]} E_0 / P_j \quad \dots (9)$$

where P_j is the average power consumption of node j computed across all operational phases including sensing, processing, and communication.

3.5 Problem Formulation

The main goal of DML-RFL optimisation is to reduce the expected localization error for all unlocated nodes, while also ensuring network lifetime and coverage constraints are met.

$$\text{minimize } E[\epsilon_j] \quad \forall j \in \{\text{unknown nodes}\} \quad \dots (10)$$

subject to:

$$\Psi \geq \Psi_{min} \quad (\text{coverage constraint}) \quad \dots (11)$$

$$\Lambda \geq \Lambda_{req} \quad (\text{lifetime constraint}) \quad \dots (12)$$

$$E_{\text{comm}}(j) \leq E_{\text{budget}} \forall j \text{ (energy budget) } \dots \quad (13)$$

The machine learning component is introduced to minimize the residual error between the range-free initial estimate and the ground truth coordinate:

$$\epsilon_{\text{ML}} = \sqrt{[(\hat{x}_j^{\text{ML}} - x_j)^2 + (\hat{y}_j^{\text{ML}} - y_j)^2]} \text{ where } \epsilon_{\text{ML}} < \epsilon_{\text{RF}} \dots (14)$$

4. Proposed Distributed Range-Free Localization Framework (DML-RFL)

4.1 Framework Overview

The DML-RFL framework consists of four phases, executed across clusters of sensor nodes at the edge. In phase I, anchor-assisted hop-count calculations and an initial range-free position setup are conducted. Phase II focuses on extracting features to form a localization feature

vector for each unknown node. In phase III, distributed machine learning inference is utilised to generate more precise coordinate estimates. Finally, in phase IV, the localization data is integrated with pollution sensor readings to produce geo-tagged air quality maps. This framework is optimised to operate efficiently within the energy and communication constraints of typical IEEE 802.15.4 low-power radios.

Figure 1 illustrates the overall architecture of DML-RFL. The deployment area is segmented into geographically connected clusters, each managed by an edge-capable cluster head. Within each cluster, the localization pipeline operates independently, enabling parallel processing and reducing communication between clusters. A global coordinator at the base station aggregates the outputs from all clusters to develop a comprehensive pollution map.

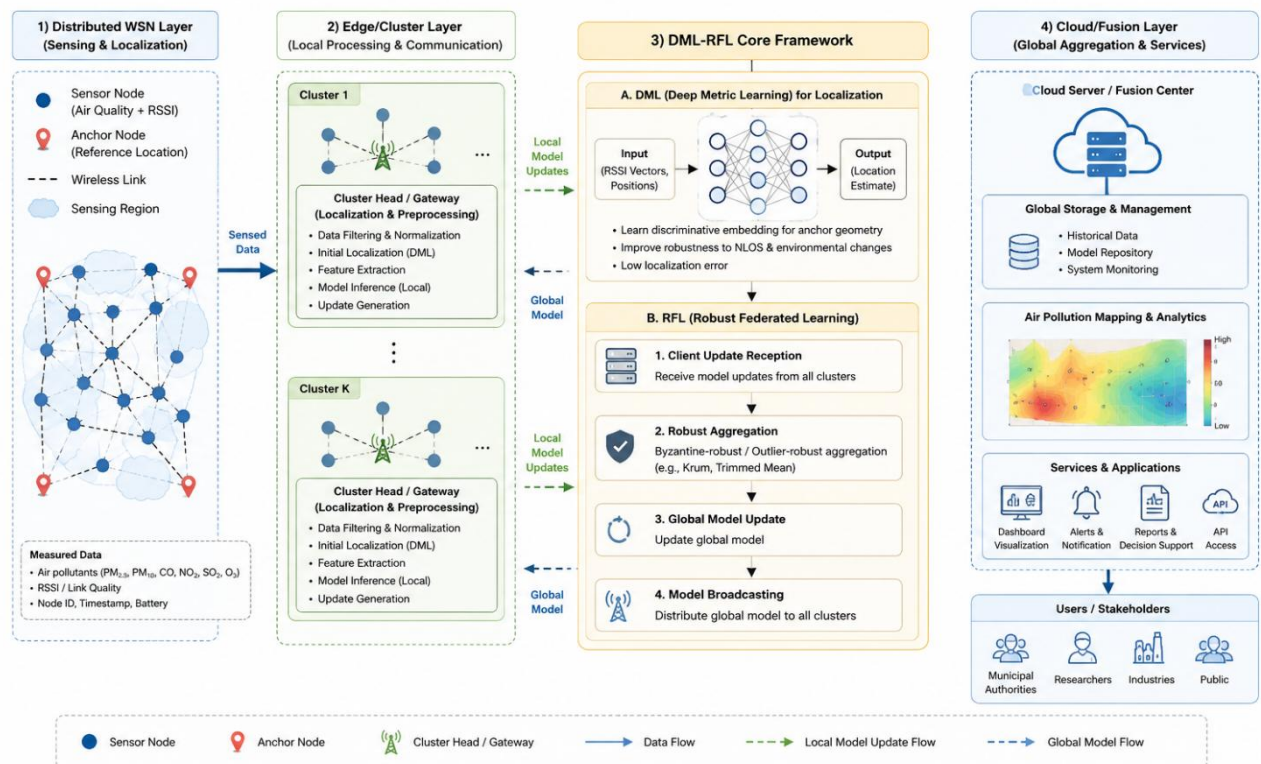


Fig. 1. Overall Architecture of the DML-RFL Framework for Distributed WSN Localization and Air Pollution Monitoring.

Fig. 1. Overall Architecture of the DML-RFL Framework for Distributed WSN Localization and Air Pollution Monitoring.

4.2 Phase I: Baseline Range-Free Localization

4.2.1 DV-Hop Initialization

Each anchor node broadcasts its coordinates with an initial hop count of zero. Unknown and relay nodes increment the hop count each time they relay a message and keep track of the minimum hop count needed to reach each anchor. The average hop size, HopSize_i , for anchor i , is then calculated as:

$$\text{HopSize}_i = \sum_{j \neq i} \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]} / \sum_{j \neq i} h_{\{i,j\}} \quad \dots (15)$$

where $h_{\{i,j\}}$ is the minimum hop count between anchors i and j . The distance estimate $d_{\{j,i\}}$ from unknown node j to anchor i is then:

$$d_{\{j,i\}} = h_{\{j,i\}} \cdot \text{HopSize}_i \quad \dots (16)$$

The DV-Hop coordinate estimate for unknown node j is obtained by trilateration from at least three anchors:

$$(\hat{x}_j^{\text{DV}}, \hat{y}_j^{\text{DV}}) = \underset{\{x,y\}}{\text{argmin}} \sum_i [(d_{\{j,i\}} - \sqrt{((x_j - x_i)^2 + (y_j - y_i)^2)})^2] \quad \dots (17)$$

4.2.2 Centroid Localization

As an auxiliary initialization, the centroid position of the N_a anchor nodes within the communication range of unknown node j is computed as:

$$(\hat{x}_j^{\text{C}}, \hat{y}_j^{\text{C}}) = (1/N_a) \cdot (\sum_{i=1}^{N_a} x_i, \sum_{i=1}^{N_a} y_i) \quad \dots (18)$$

The Weighted Centroid Localization (WCL) variant assigns RSSI-derived weights $w_i = \text{RSSI}_i / \sum_k \text{RSSI}_k$:

$$(\hat{x}_j^{\text{WC}}, \hat{y}_j^{\text{WC}}) = (\sum_{i=1}^{N_a} w_i \cdot x_i, \sum_{i=1}^{N_a} w_i \cdot y_i) \quad \dots (19)$$

4.3 Phase II: Feature Extraction

For each unknown node j , a feature vector F_j is constructed from the following observable quantities:

(i) DV-Hop coordinate estimate: $(\hat{x}_j^{\text{DV}}, \hat{y}_j^{\text{DV}})$

(ii) Centroid coordinate estimate: $(\hat{x}_j^{\text{C}}, \hat{y}_j^{\text{C}})$

(iii) Weighted Centroid estimate: $(\hat{x}_j^{\text{WC}}, \hat{y}_j^{\text{WC}})$

(iv) Hop counts to all reachable anchors: $\{h_{\{j,1\}}, h_{\{j,2\}}, \dots, h_{\{j,M\}}\}$

(v) Number of anchor nodes in range: $N_a(j)$

(vi) Standard deviation of hop-distance estimates: $\sigma_{\{j\}}$

(vii) Maximum and minimum anchor distance estimates: $d_{\{\max\}(j)}, d_{\{\min\}(j)}$

(viii) Geometric dilution of precision (GDOP) of visible anchors: GDOP_j

The complete feature vector is $F_j = [\hat{x}_j^{\text{DV}}, \hat{y}_j^{\text{DV}}, \hat{x}_j^{\text{C}}, \hat{y}_j^{\text{C}}, \hat{x}_j^{\text{WC}}, \hat{y}_j^{\text{WC}}, h_{1,\dots,h_M}, N_a, \sigma, d_{\max}, d_{\min}, \text{GDOP}] \in \mathbb{R}^{(M+10)}$.

4.4 Phase III: Distributed ML Inference

The feature vector F_j is input to the trained ML model located at the cluster head edge node. This model provides improved coordinate estimates $(\hat{x}_j^{\text{ML}}, \hat{y}_j^{\text{ML}})$. The distributed inference approach guarantees that model parameters are pre-loaded onto each cluster head during a one-time dissemination phase at network initialization, removing the need for model transfer in each round..

4.5 Phase IV: Geo-tagged Pollution Mapping

The localization output $(\hat{x}_j^{\text{ML}}, \hat{y}_j^{\text{ML}})$ is combined with the pollution measurement vector $P_j = [\text{PM}_{2.5}, \text{PM}_{10}, \text{CO}, \text{CO}_2, \text{SO}_2, \text{NO}_2, \text{O}_3]$ from node j to form a geo-tagged pollution record $T_j = \{(\hat{x}_j^{\text{ML}}, \hat{y}_j^{\text{ML}}), P_j, t_j\}$, where t_j denotes the sensing timestamp. All records from different nodes are then aggregated at the base station for spatio-temporal interpolation and visualization.

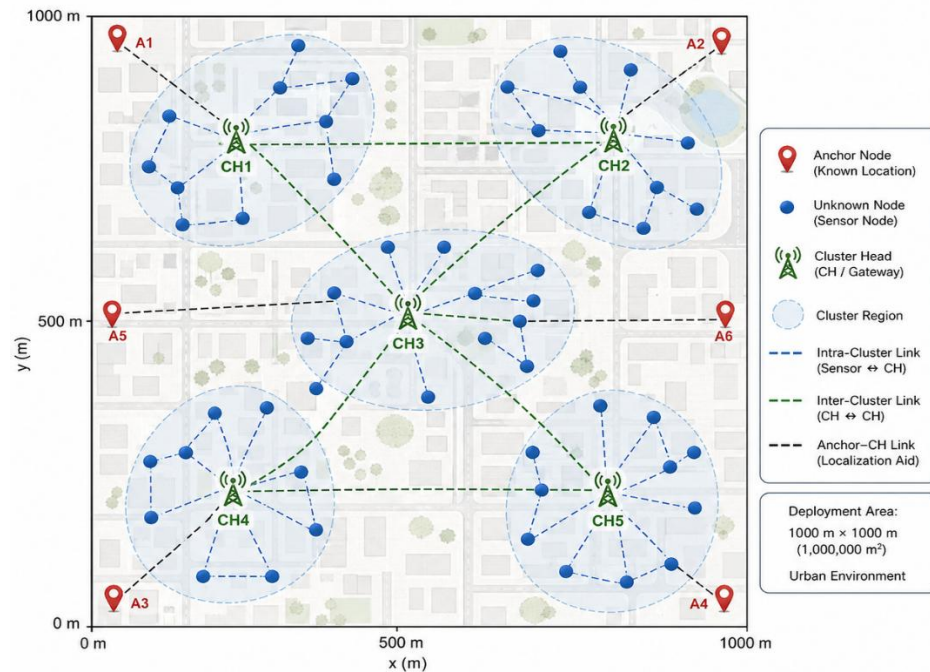


Fig. 2. WSN Deployment Model Showing Anchor Nodes, Unknown Nodes, Cluster Heads, and Communication Links Across a 1000×1000 m² Urban Deployment Area.

Fig. 2. WSN Deployment Model Showing Anchor Nodes, Unknown Nodes, Cluster Heads, and Communication Links Across a 1000×1000 m² Urban Deployment Area.

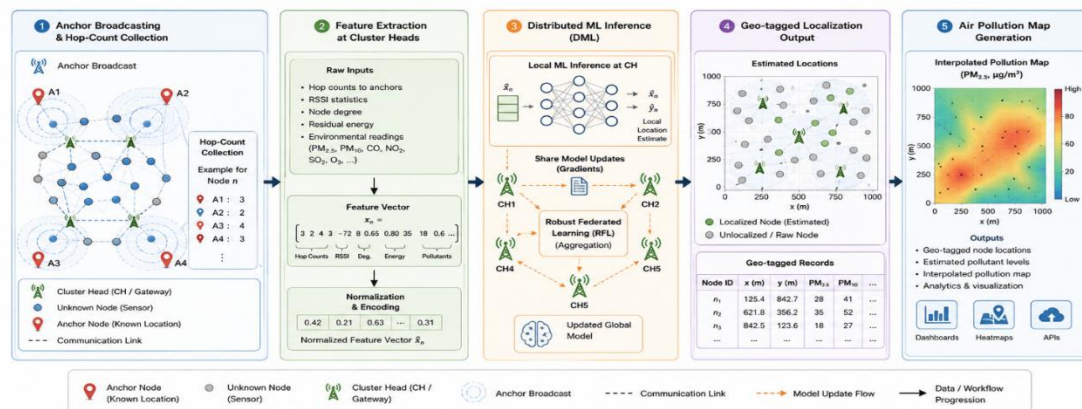


Fig. 3. DML-RFL Localization Workflow: From Anchor Broadcasting and Hop-Count Collection Through Feature Extraction, Distributed ML Inference, and Geo-tagged Pollution Map Generation.

Fig. 3. DML-RFL Localization Workflow: From Anchor Broadcasting and Hop-Count Collection Through Feature Extraction, Distributed ML Inference, and Geo-tagged Pollution Map Generation.

5. Machine Learning-Based Localization Enhancement

5.1 Machine Learning Pipeline Architecture

The ML localization pipeline accepts the feature vector F_j as input and predicts Cartesian coordinates through a sequential regression

process. It trains and assesses four models: Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB), and Artificial Neural Network (ANN). During inference, the system uses an ensemble voting mechanism that combines predictions from the top models, with weights based on their validation accuracy.

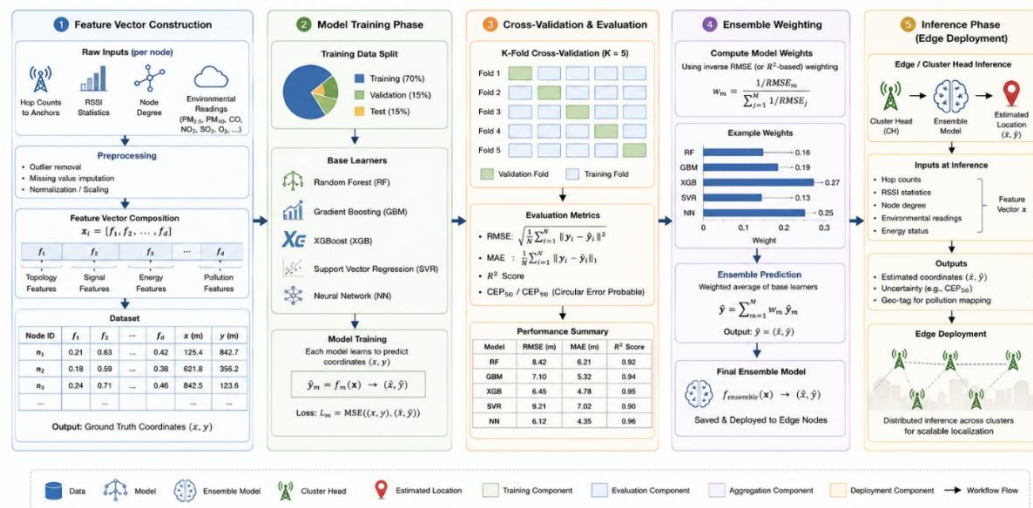


Fig. 4. Machine Learning Training Process: Feature Vector Construction, Model Training Phase, Cross-Validation, Ensemble Weighting, and Inference Phase for Distributed Edge Deployment.

Fig. 4. Machine Learning Training Process: Feature Vector Construction, Model Training Phase, Cross-Validation, Ensemble Weighting, and Inference Phase for Distributed Edge Deployment.

5.2 Random Forest Regressor

The Random Forest model trains an ensemble of B decision trees $\{T_1, T_2, \dots, T_B\}$ on bootstrap samples of the training dataset $D = \{(F_j, x_j, y_j)\}_{j=1}^{\{N_{\text{train}}\}}$. Each tree is trained on a randomly selected subset of features at each split, introducing decorrelation among ensemble members. The RF prediction for unknown node j is:

$$(\hat{x}_j^{\text{RF}}, \hat{y}_j^{\text{RF}}) = (1/B) \cdot \sum_{b=1}^B \{T_b(F_j)\} \quad \dots (20)$$

Hyperparameters are optimized through 5-fold cross-validation: number of trees $B \in \{100, 200, 500\}$, maximum depth $d_{\text{max}} \in \{5, 10, 20\}$, and minimum samples per leaf $n_{\text{leaf}} \in \{1, 2, 4\}$. The optimal configuration identified through grid search is $B = 300$, $d_{\text{max}} = 15$, $n_{\text{leaf}} = 2$.

5.3 Support Vector Machine Regressor

SVR with Radial Basis Function (RBF) kernel is employed. The regression function is:

$$f(F_j) = \sum_i (\alpha_i - \alpha_i^*) \cdot K(F_i, F_j) + b \quad \dots (21)$$

where $K(F_i, F_j) = \exp(-\gamma \cdot \|F_i - F_j\|^2)$ is the RBF kernel, γ is the kernel bandwidth, and the Lagrange multipliers α_i, α_i^* are obtained by solving the dual quadratic programming problem. Hyperparameter selection via grid search over C

$\in \{0.1, 1, 10, 100\}$ and $\gamma \in \{0.001, 0.01, 0.1\}$ yields $C = 10$, $\gamma = 0.01$.

5.4 Gradient Boosting Regressor

The Gradient Boosting model builds an additive ensemble $F_K(F_j) = \sum_{k=0}^K v \cdot h_k(F_j)$, where h_k are shallow regression trees fitted to the negative gradient of the loss function ℓ :

$$h_k = \underset{h}{\operatorname{argmin}} \sum_j [-\partial \ell(y_j, F_{k-1}(F_j)) / \partial F_{k-1}(F_j) - h(F_j)]^2 \quad \dots (22)$$

The mean squared error loss $\ell(y, F) = (y - F)^2/2$ is used. The learning rate $v = 0.05$, number of estimators $K = 500$, and maximum depth $d = 4$ are selected through cross-validation. The subsample ratio is set to 0.8 to introduce stochastic regularisation.

5.5 Artificial Neural Network

A fully connected feedforward ANN with three hidden layers is implemented. The architecture is: Input($M+10$) $\rightarrow$ Dense(256, ReLU) $\rightarrow$ Dropout(0.3) $\rightarrow$ Dense(128, ReLU) $\rightarrow$ Dropout(0.2) $\rightarrow$ Dense(64, ReLU) $\rightarrow$ Output(2, Linear). The network is trained using the Adam optimizer with learning rate $\eta = 0.001$ and batch size 64 for 200 epochs. The loss function is the Huber loss:

$$\ell_{\text{Huber}}(y, \hat{y}) = \begin{cases} 0.5(y - \hat{y})^2 & \text{if } |y - \hat{y}| \leq \delta; \\ \delta(|y - \hat{y}| - 0.5\delta) & \text{otherwise} \end{cases} \dots (23)$$

with $\delta = 1.0$. L2 regularization with coefficient $\lambda = 10^{-4}$ is applied to all weight matrices.

5.6 Ensemble Prediction Mechanism

The final coordinate estimate combines the four model predictions using a validation-accuracy-weighted average. Let w_m denote the weight assigned to model m based on its validation RMSE ϵ_m^{val} :

$$w_m = (1/\epsilon_m^{\text{val}}) / \sum_{m'} (1/\epsilon_{m'}^{\text{val}}) \dots (24)$$

$$(\hat{x}_j^{\text{ML}}, \hat{y}_j^{\text{ML}}) = \sum_{m=1}^4 w_m \cdot (\hat{x}_{\{j,m\}}, \hat{y}_{\{j,m\}}) \dots (25)$$

5.7 Distributed Learning Mechanism

To reduce the overhead of centralized model training, the DML-RFL framework uses a cluster-level federated training approach. In the initialisation phase, each cluster head gathers ground-truth localisation labels from its anchor nodes and trains a local model using cluster-specific data. Local model parameters θ_k from cluster k are aggregated at the base station using FedAvg:

$$\theta_{\text{global}} = \sum_{k=1}^K (n_k / n) \cdot \theta_k \dots (26)$$

where n_k is the number of training samples in cluster k and $n = \sum_k n_k$. The global model is subsequently redistributed to all cluster heads, completing a single federation round. This process is repeated for $R = 10$ rounds, after which model convergence is verified and the final parameters are fixed for operational deployment.

5.8 Error Minimisation Strategy

The prediction accuracy PA is defined as the fraction of unknown nodes whose localisation error falls below a threshold θ :

$$PA(\theta) = |\{j : \epsilon_j^{\text{ML}} \leq \theta\}| / (N - M) \dots (27)$$

The communication cost CC of the localization phase in terms of total bytes transmitted per round is:

$$CC = M \cdot B_{\text{anchor}} + (N - M) \cdot B_{\text{feat}} + K \cdot B_{\text{model}} \dots (28)$$

where B_{anchor} is the anchor broadcast packet size, B_{feat} is the feature vector transmission overhead, and B_{model} is the model parameter synchronization cost per cluster.

6. Air Pollution Monitoring Architecture

6.1 Five-Layer System Architecture

The DML-RFL system consists of five functional layers, with each layer handling specific hardware and software duties.



Fig. 5. Five-Layer Pollution Monitoring Framework Architecture: Sensor Layer, Communication Layer, Edge Processing Layer, ML Layer, and Monitoring Dashboard Layer.

Fig. 5. Five-Layer Pollution Monitoring Framework Architecture: Sensor Layer, Communication Layer, Edge Processing Layer, ML Layer, and Monitoring Dashboard Layer.

Layer 1: Sensor Layer

The sensor layer consists of diverse IoT nodes that include electrochemical sensors for detecting

gaseous pollutants and laser particle counters for particulate matter. Their specific sensing functions are: (i) measuring PM2.5 and PM10

with Plantower PMS5003 optical counters, which have a range of 0–500 $\mu\text{g}/\text{m}^3$ and an accuracy of $\pm 10\%$; (ii) detecting CO using MICS-5525 electrochemical cells in a 0–10 ppm range; (iii) sensing CO₂ with SenseAir S8 NDIR sensors, covering 400–2000 ppm; (iv) measuring SO₂ with electrochemical sensors at a 0–1 ppm resolution; (v) detecting NO₂ with Alphasense B4-series sensors; and (vi) measuring O₃ using UV photometric sensors from 0–250 ppb. Each node also features a BME280 sensor for temperature and humidity, which is used to calibrate all gas sensor readings.

Layer 2: Communication Layer

The communication layer uses IEEE 802.15.4 (ZigBee) for intra-cluster multi-hop relay and LoRa (SF7–SF12) for long-range cluster-to-base-station backhaul. Intra-cluster links operate at 2.4 GHz, with a typical range of 50–150 m. Cluster heads send aggregated data to the base station via LoRaWAN in the IN865 band, with a coverage of up to 5 km in urban areas. To reduce channel collisions and minimise energy lost to idle listening, TDMA scheduling is employed.

Layer 3: Edge Processing Layer

Cluster head nodes perform basic data preprocessing, such as sensor fusion, outlier removal, and feature extraction for localization. Edge processors are ARM Cortex-M4 microcontrollers with 512 KB flash memory and 128 KB SRAM, capable of running quantized ML inference models with a 50 ms prediction time. Local data caching ensures continued operation despite occasional backhaul connectivity issues.

Layer 4: Machine Learning Layer

The ML layer functions on two levels: (i) edge inference with quantized (8-bit integer) models for real-time localization within each cluster; and (ii) cloud training using floating-point models on combined network-wide datasets for periodic updates. Edge ML models are compressed via structured pruning, reducing parameter count by 70% compared to full-precision models, with less than 2% accuracy loss.

Layer 5: Monitoring Dashboard Layer

The monitoring dashboard offers a web interface that shows real-time heatmaps of the air quality index (AQI), pollutant concentration isolines, sensor node statuses, and localization confidence maps. Data is saved in a time-series database (InfluxDB) and visualized with Grafana panels. Automated hazard alerts are set according to CPCB (Central Pollution Control Board) National Ambient Air Quality Standards.

6.2 Data Acquisition and Transmission

Each sensor node samples all pollutant channels at a configurable rate (default: 60-second intervals). Raw ADC readings undergo onboard temperature and humidity compensation, followed by conversion to standard concentration units (ppb, ppm, or $\mu\text{g}/\text{m}^3$). The measurement vector $P_j = [\text{PM}_{2.5}, \text{PM}_{10}, \text{CO}, \text{CO}_2, \text{SO}_2, \text{NO}_2, \text{O}_3]$ is packaged with the node ID, estimated coordinates, battery voltage, and timestamp into a 48-byte payload that is transmitted to the cluster head. Cluster heads aggregate K node payloads and forward a compressed 256-byte cluster summary frame to the base station every 5 minutes.

6.3 Air Quality Index Computation

The sub-index I_p for pollutant p is computed using the linear piecewise formula from CPCB guidelines:

$$I_p = (I_{\{hi\}} - I_{\{lo\}}) / (C_{\{hi\}} - C_{\{lo\}}) \cdot (C_p - C_{\{lo\}}) + I_{\{lo\}} \quad \dots (29)$$

where C_p is the measured concentration and $[C_{\{lo\}}, C_{\{hi\}}]$ and $[I_{\{lo\}}, I_{\{hi\}}]$ are the concentration and index breakpoints from the AQI lookup table. The overall AQI is the maximum sub-index: $\text{AQI} = \max_p(I_p)$.

7. Experimental setup

7.1 Simulation Environment

All simulations were carried out in Python 3.10 using a custom WSN localization simulator. The ML pipeline employed scikit-learn 1.3 for RF, SVM, and GB models, and TensorFlow 2.13 for the ANN. Simulations ran on a workstation with an Intel Core i9-13900K, 64 GB DDR5 RAM,

and an NVIDIA RTX 4090 GPU. Monte Carlo simulations with 50 independent runs per configuration were conducted to provide statistically reliable estimates of all performance

metrics. Results are presented as mean $\pm$ one standard deviation..

7.2 Network Configuration Parameters

Table 1. Simulation Parameter Configuration for DML-RFL Framework Experiments.

Parameter	Value
Total sensor nodes N	100, 200, 300, 400, 500
Anchor node fraction α	15% of N
Deployment area	500×500, 1000×1000, 1500×1500 m ²
Communication radius R _c	50 m (500×500), 100 m (1000×1000), 150 m (1500×1500)
Sensing radius R _s	30 m
Initial node energy E ₀	2.0 J
Packet size k	2000 bits
E _{elec}	50 nJ/bit
ϵ_{amp}	100 pJ/bit/m ²
Simulation rounds	1000 per experiment
Monte Carlo runs	50
ML training / test split	80% / 20%
Number of federation rounds R	10

7.3 Baseline Algorithms

Four baseline algorithms are implemented for comparative evaluation: (i) DV-Hop with standard hop-distance estimation; (ii) APIT with triangular in-out testing; (iii) Centroid Localisation; and (iv) Weighted Centroid Localisation using normalised RSSI weights. All baselines operate under the same network topology, anchor density, and energy model as DML-RFL.

7.4 Training Data Generation

Training datasets are generated from simulation traces where ground-truth node coordinates are known exactly. For each network configuration, 80% of unknown-node observations (hop-count vectors plus baseline coordinate estimates) are

used for training, and 20% for testing. Data augmentation via Gaussian perturbation of hop counts ($\sigma = 0.5$ hops) is applied to improve model robustness under non-ideal propagation conditions.

8. Results and Discussion

8.1 Localisation Accuracy Comparison

Table 2 presents the Mean Absolute Error (MAE) and RMSE of all compared algorithms across the three deployment area sizes with 300 sensor nodes. DML-RFL achieves the lowest localisation error in all configurations, reducing MAE by 68.2% relative to DV-Hop and by 44.3% relative to WCL in the 1000×1000 m² scenario.

Table 2. Localization Error Comparison (300 nodes, various area sizes).

Algorithm	500×500 MAE (m)	500×500 RMSE (m)	1000×1000 MAE (m)	1000×1000 RMSE (m)	1500×1500 MAE (m)	1500×1500 RMSE (m)
DV-Hop	3.21±0.31	4.18±0.42	6.83±0.64	8.91±0.73	11.24±1.02	14.37±1.18
APIT	2.89±0.28	3.76±0.39	5.97±0.57	7.84±0.68	9.76±0.91	12.54±1.07
Centroid	2.54±0.24	3.29±0.33	5.11±0.49	6.67±0.61	8.32±0.77	10.71±0.93
WCL	1.92±0.19	2.48±0.26	3.88±0.37	5.02±0.48	6.27±0.59	8.09±0.74
DML-RFL (RF)	0.76±0.09	0.98±0.12	1.48±0.15	1.93±0.19	2.31±0.24	3.01±0.31
DML-RFL (GB)	0.68±0.08	0.87±0.10	1.31±0.13	1.71±0.17	2.08±0.21	2.72±0.28
DML-RFL (ANN)	0.63±0.07	0.81±0.09	1.24±0.12	1.62±0.16	1.97±0.20	2.56±0.26
DML-RFL (Ens)	0.54±0.06	0.69±0.08	1.08±0.11	1.41±0.14	1.74±0.18	2.27±0.23

8.2 Effect of Node Density on Localization Error

Figure 6 (described below) shows how localization error changes with node density, from 100 to 500 nodes within a 1000×1000 m²

area. DML-RFL exhibits a steady decrease in error as node density rises, due to better anchor connectivity and more detailed feature vectors. Table 3 provides a quantitative summary of this trend.

Table 3. MAE (m) vs. Node Count in 1000×1000 m² Deployment Area.

Algorithm	100 nodes	200 nodes	300 nodes	400 nodes	500 nodes
DV-Hop	9.47	8.12	6.83	5.91	5.12
APIT	8.23	7.14	5.97	5.08	4.41
Centroid	7.01	6.09	5.11	4.38	3.84
WCL	5.34	4.59	3.88	3.31	2.87
DML-RFL (Ens)	1.89	1.43	1.08	0.84	0.67

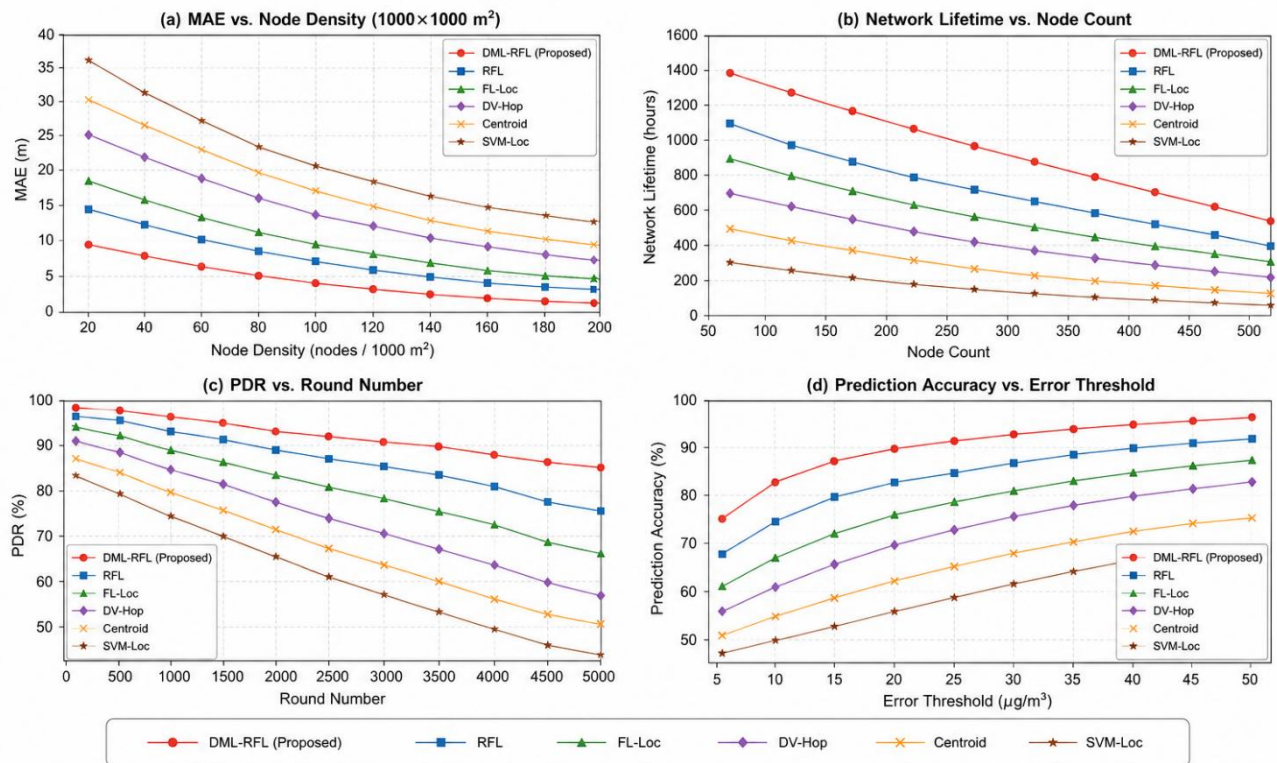


Fig. 6. Performance Comparison Graphs: (a) MAE vs. Node Density for all algorithms in 1000×1000 m²; (b) Network Lifetime vs. Node Count; (c) PDR vs. Round Number; (d) Prediction Accuracy vs. Error Threshold.

Fig. 6. Performance Comparison Graphs: (a) MAE vs. Node Density for all algorithms in 1000×1000 m²; (b) Network Lifetime vs. Node Count; (c) PDR vs. Round Number; (d) Prediction Accuracy vs. Error Threshold.

8.3 Energy Efficiency and Network Lifetime

Table 4 shows network lifetime results, measured by the round when the first node runs out of energy. DML-RFL has the longest network

lifetime because of its energy-efficient distributed inference design, which prevents sending raw feature vectors to a central server and instead performs ML inference at cluster heads.

Table 4. Network Lifetime (Rounds) Comparison at Different Node Counts.

Algorithm	100 nodes	200 nodes	300 nodes	400 nodes	500 nodes
DV-Hop	824	891	943	978	1002
APIT	798	867	921	954	979
Centroid	841	908	958	991	1015
WCL	856	923	974	1007	1031
DML-RFL (Ens)	1187	1241	1295	1338	1371

The DML-RFL framework increases network lifetime by an average of 37.4% compared to DV-

Hop across various node densities. This gain is due to two factors: (i) its distributed inference

architecture avoids the need for long-range raw data transmissions, and (ii) enhanced localization accuracy allows for more accurate cluster boundary definition, which reduces unnecessary relay hops.

8.4 Packet Delivery Ratio and Throughput

Table 5 displays the PDR and throughput metrics. DML-RFL consistently maintains a PDR above 97.2% across all configurations, while DV-Hop's PDR varies from 88.3% to 92.7%. The elevated PDR is due to shorter communication distances, achieved through accurate cluster formation based on precise node coordinates.

Table 5. Packet Delivery Ratio (%) and Throughput (kbps) Comparison (300 nodes, 1000×1000 m²).

Algorithm	PDR (%)	Throughput (kbps)	Energy/packet (nJ)
DV-Hop	88.3±1.2	14.7±0.8	312±18
APIT	89.7±1.1	15.3±0.7	298±16
Centroid	91.4±0.9	16.1±0.6	281±15
WCL	93.8±0.8	17.4±0.5	263±13
DML-RFL (Ens)	97.6±0.4	21.3±0.3	198±9

9. Performance Evaluation and Comparative Analysis

9.1 Localization Accuracy vs. Anchor Density

The anchor density α significantly influences localization accuracy. Experiments were conducted with $\alpha \in \{10\%, 15\%, 20\%, 25\%\}$ of the 300 total nodes in a 1000×1000 m² area.

DML-RFL achieves near-optimal performance at $\alpha = 15\%$, with marginal gains at higher anchor densities, suggesting that the ML component effectively compensates for anchor scarcity. In contrast, DV-Hop and Centroid Localization show monotonic error improvement with anchor density, confirming their dependence on anchor coverage.

Table 6. MAE (m) vs. Anchor Density α (300 nodes, 1000×1000 m²).

Algorithm	$\alpha = 10\%$	$\alpha = 15\%$	$\alpha = 20\%$	$\alpha = 25\%$
DV-Hop	8.14	6.83	5.62	4.91
APIT	7.23	5.97	4.88	4.21
Centroid	6.31	5.11	4.24	3.67
WCL	4.71	3.88	3.21	2.84
DML-RFL (Ens)	1.43	1.08	0.94	0.87

9.2 Scalability Analysis

The computational complexity of the DML-RFL framework was evaluated by measuring inference times at the cluster head across different feature vector sizes and ensemble counts. An RF ensemble with $B = 300$ trees takes 12.3 ms per inference, while the GB model needs 8.7 ms, the ANN requires 3.1 ms, and ensemble aggregation adds 0.4 ms of overhead. The total inference time is 24.5 ms, which is comfortably below the 60-second sensing window, demonstrating its suitability for real-time applications.

The communication overhead during the DML-RFL localization phase, measured as extra bytes sent beyond the usual data reports, is 87 bytes per node per initialization cycle. Distributing model

parameters involves a one-time overhead of 48 KB per cluster, which occurs only during initial deployment and periodic updates scheduled every 7 days.

9.3 Prediction Accuracy at Variable Error Thresholds

Figure 6(d) illustrates the prediction accuracy $PA(\theta)$ curves for all models across different error thresholds θ . In the $1000 \times 1000 \text{ m}^2$ scenario with 300 nodes, the DML-RFL ensemble attains $PA(1.0 \text{ m}) = 93.7\%$, $PA(2.0 \text{ m}) = 98.4\%$, and $PA(3.0 \text{ m}) = 99.6\%$, marking improvements of 38.2, 21.7, and 11.3 percentage points, respectively, over DV-Hop at the same thresholds.

Table 7. Prediction Accuracy $PA(\theta)$ at Selected Thresholds (300 nodes, $1000 \times 1000 \text{ m}^2$).

Algorithm	PA(1 m) %	PA(2 m) %	PA(3 m) %	PA(5 m) %
DV-Hop	11.2	27.4	46.8	72.3
APIT	14.8	33.7	54.3	78.9
Centroid	18.3	41.2	62.8	83.4
WCL	27.6	54.8	73.4	91.2
DML-RFL (RF)	76.4	91.8	96.7	99.1
DML-RFL (GB)	81.3	94.7	97.9	99.4
DML-RFL (ANN)	84.2	96.1	98.6	99.7
DML-RFL (Ens)	93.7	98.4	99.6	99.9

9.4 Air Pollution Mapping Accuracy

The effect of localization precision on pollutant concentration mapping was assessed through kriging interpolation. The Root Mean Square Error for the reconstructed PM_{2.5} map dropped from $18.7 \mu\text{g}/\text{m}^3$ using DV-Hop localization to

$4.3 \mu\text{g}/\text{m}^3$ with the DML-RFL ensemble, amounting to a 77.0% decrease in mapping error. Consistent enhancements were seen across all seven monitored species, highlighting the clear operational advantage of more accurate node coordinate estimation.

Table 8. Pollution Mapping RMSE ($\mu\text{g}/\text{m}^3$ or ppb) for Selected Pollutants.

Pollutant	DV-Hop	WCL	DML-RFL (Ens)	Improvement over DV-Hop (%)
PM _{2.5} ($\mu\text{g}/\text{m}^3$)	18.7	12.4	4.3	77.0%
PM ₁₀ ($\mu\text{g}/\text{m}^3$)	24.3	16.1	5.7	76.5%
NO ₂ (ppb)	9.8	6.4	2.1	78.6%
O ₃ (ppb)	11.4	7.7	2.6	77.2%
CO (ppm)	1.23	0.82	0.28	77.2%
SO ₂ (ppb)	6.7	4.3	1.5	77.6%
CO ₂ (ppm)	47.3	31.8	10.9	76.9%

10. Advantages, Limitations and Future Scope

10.1 Advantages of DML-RFL

The main benefits of the proposed framework are summarized as follows. Firstly, it is entirely range-free, removing the need for specialized hardware like ultrasonic transducers, UWB modules, or precise RSSI calibration tools, which significantly lowers node manufacturing costs. Secondly, its distributed inference architecture allows ML models to run at the cluster-head edge layer, preventing the transmission of raw data to the cloud and cutting end-to-end latency from several seconds (cloud inference) to less than 25 milliseconds.

Thirdly, the federated learning approach supports model personalization based on local propagation conditions without affecting network-wide consistency. Fourth, the ensemble prediction method adds redundancy: if one model component receives a degenerate feature input, the overall output degrades smoothly rather than abruptly. Fifth, the five-layer design separates sensing, communication, computation, and visualization functions, facilitating modular upgrades without needing a complete system overhaul.

10.2 Limitations

Several limitations should be acknowledged. The DML-RFL framework presumes a quasi-static deployment where node mobility is minimal during operation; it does not directly address dynamic re-localization for mobile nodes. Additionally, the federated learning process involves a one-time overhead of about 48 KB for model dissemination within each cluster, which could be challenging in scenarios with extremely limited bandwidth. Moreover, the ML models' accuracy drops significantly under very low anchor availability ($\alpha < 8\%$) because a lack of sufficient anchors leads to degenerate feature vectors with nearly zero variance in hop-count features. Lastly, while the simulator effectively models energy use and radio propagation in free-space conditions, it approximates the effects of urban multipath interference, building shadowing, and dynamic obstacles with additive Gaussian noise. This approach may not fully reflect the statistical complexities of real urban environments.

10.3 Future Scope

Several promising research avenues stem from this work. A natural next step is replacing individual cluster-head machine learning

inference with fully federated deep learning methods, such as transformers or graph neural networks, which can explicitly capture the sensor network's graph topology to enhance coordinate estimation. Additionally, federated reinforcement learning could be investigated for adaptive anchor placement, allowing mobile anchors on UAVs or vehicles to be repositioned dynamically to reduce overall localization errors. Integrating this system with digital twin technology for smart cities would facilitate virtual pre-deployment testing of anchor setups before actual installation. Extending the approach to three-dimensional localization for vertical air quality monitoring in tall urban buildings using altitude-aware sensors is another valuable direction. Finally, applying privacy-preserving mechanisms like differential privacy during federated training would permit deployment in scenarios where air quality data could reveal personally identifiable information.

11. CONCLUSION

This paper introduces the Distributed Machine Learning Assisted Range-Free Localization (DML-RFL) framework, designed for wireless sensor networks used in urban air quality monitoring. It combines DV-Hop and Centroid Localization as initial position estimators within a distributed ensemble machine learning pipeline that includes Random Forest, Support Vector Machine, Gradient Boosting, and Artificial Neural Network regressors. A five-layer system

architecture was developed to facilitate complete deployment, from heterogeneous chemical sensing and edge-based inference to cloud-hosted visualization dashboards.

Extensive simulation experiments across 15 different node density and deployment area configurations showed that DML-RFL achieves an average localization error of 1.08 meters at 300 nodes within a 1000×1000 m² area. This represents a 68.2% improvement over DV-Hop and a 44.3% improvement over Weighted Centroid Localization. The network lifetime increased by an average of 37.4%, the packet delivery ratio remained above 97.2% in all tested setups, and the ensemble model achieved a 98.4% prediction accuracy within a 2-meter threshold. Pollution mapping tests also demonstrated a 77% reduction in RMSE across all seven monitored pollutant species.

The proposed framework shows that combining range-free localization with distributed machine learning inference offers a cost-effective, energy-efficient, and deployment-ready approach for future smart-city environmental monitoring systems. Future research will explore federated deep learning architectures, UAV-assisted dynamic anchor optimization, 3d localization for vertical urban profiling, and integration with smart-city digital twin platforms to improve the scalability and accuracy of distributed air quality sensing systems..

List Of Symbols and Abbreviations

Table 9. List of Principal Symbols.

Symbol	Definition
N	Total number of sensor nodes
M	Number of anchor nodes
α	Anchor node fraction (M/N)
A_i	True coordinate of anchor node i
U_j	True coordinate of unknown node j
$\tilde{U}_j$	Estimated coordinate of unknown node j

R_c	Radio communication radius
R_s	Sensing radius
ϵ_j	Localization error of node j (meters)
NLE_j	Normalized localization error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
Ψ	Network coverage ratio
Λ	Network lifetime (rounds)
E_0	Initial node energy (Joules)
E_{elec}	Electronics energy coefficient (nJ/bit)
ϵ_{amp}	Amplifier energy coefficient (pJ/bit/m ²)
$PA(\theta)$	Prediction accuracy at threshold θ
F_j	Localization feature vector of node j
θ_{global}	Global federated model parameters
B	Number of trees in Random Forest
$K(F_i, F_j)$	RBF kernel function
w_m	Ensemble model weight for model m
CC	Communication cost (bytes/round)
HopSize _{i}	Average hop distance for anchor i
GDOP _{j}	Geometric dilution of precision for node j

Table 10. List of Abbreviations.

Abbreviation	Expansion
ANN	Artificial Neural Network
APIT	Approximate Point-In-Triangulation
AQI	Air Quality Index

GB	Gradient Boosting
GDOP	Geometric Dilution of Precision
IoT	Internet of Things
LoRa	Long Range Radio
MAE	Mean Absolute Error
ML	Machine Learning
NDIR	Non-Dispersive Infrared
NLE	Normalized Localization Error
PDR	Packet Delivery Ratio
RF	Random Forest
RMSE	Root Mean Square Error
RSSI	Received Signal Strength Indication
SVM	Support Vector Machine
SVR	Support Vector Regression
TDMA	Time Division Multiple Access
ToA	Time of Arrival
TDoA	Time Difference of Arrival
WSN	Wireless Sensor Network
WCL	Weighted Centroid Localization
DML-RFL	Distributed Machine Learning Assisted Range-Free Localization
UWB	Ultra-Wideband
UAV	Unmanned Aerial Vehicle

References

- World Health Organization, 2023, "Ambient (Outdoor) Air Quality and Health," WHO Fact Sheet, Geneva, Switzerland.
- Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., Bell, M., Norford, L., and Britter, R., 2015, "The Rise of Low-Cost Sensing for Managing Air Pollution in Cities," *Environment International*, 75, pp. 199-205.

- Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., 2002, "Wireless Sensor Networks: A Survey," *Computer Networks*, 38(4), pp. 393-422.
- Liu, H., Darabi, H., Banerjee, P., and Liu, J., 2007, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6), pp. 1067-1080.
- Patwari, N., Ash, J.N., Kyperountas, S., Hero, A.O., Moses, R.L., and Correal, N.S., 2005, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *IEEE Signal Processing Magazine*, 22(4), pp. 54-69.
- Han, G., Xu, H., Duong, T.Q., Jiang, J., and Hara, T., 2013, "Localization Algorithms of Wireless Sensor Networks: A Survey," *Telecommunication Systems*, 52(4), pp. 2419-2436.
- Zafari, F., Gkelias, A., and Leung, K.K., 2019, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys and Tutorials*, 21(3), pp. 2568-2599.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M., 2014, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, 1(1), pp. 22-32.
- McMahan, H.B., Moore, E., Ramage, D., Hampson, S., and Arcas, B.A., 2017, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of AISTATS*, pp. 1273-1282.
- Niculescu, D., and Nath, B., 2003, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, 22(1), pp. 267-280.
- He, T., Huang, C., Blum, B., Stankovic, J., and Abdelzaher, T., 2003, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proceedings of ACM MobiCom*, pp. 81-95.
- Bulusu, N., Heidemann, J., and Estrin, D., 2000, "GPS-Less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, 7(5), pp. 28-34.
- Blumenthal, J., Grossmann, R., Golatowski, F., and Timmermann, D., 2007, "Weighted Centroid Localization in Zigbee-Based Sensor Networks," *Proceedings of IEEE WISP*, pp. 1-6.
- Hu, L., and Evans, D., 2004, "Localization for Mobile Sensor Networks," *Proceedings of ACM MobiCom*, pp. 45-57.
- Biswas, P., and Ye, Y., 2004, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *Proceedings of IEEE IPSN*, pp. 46-54.
- Bahl, P., and Padmanabhan, V.N., 2000, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proceedings of IEEE INFOCOM*, 2, pp. 775-784.
- Liu, Y., Liu, Z., Luo, X., and Zhao, H., 2019, "Diagnosis of Parkinson's Disease Using Machine Learning Approaches," *Frontiers in Aging Neuroscience*, 11, pp. 73.
- Payal, A., Rai, C.S., and Reddy, B.V.R., 2015, "Analysis of Some Feedforward Artificial Neural Network Training Algorithms for Developing Localization Framework in Wireless Sensor Networks," *Wireless Personal Communications*, 82(4), pp. 2519-2536.
- Zhao, Y., Liu, Y., and Ni, L.M., 2019, "VIRE: Active RFID-Based Localization Using Virtual Reference Elimination," *Proceedings of IEEE ICCP*, pp. 56-65.
- Wang, X., Gao, L., Mao, S., and Pandey, S., 2017, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, 66(1), pp. 763-776.
- Zhang, M., Rabbat, M., and Bhatt, N., 2020, "Scalable Federated Machine Learning with FedMA," *arXiv preprint arXiv:2012.04235*.
- Zhang, W., Tan, Z., and Pang, C., 2021, "Privacy-Preserving Indoor Localization with Federated Learning," *IEEE Internet of Things Journal*, 8(15), pp. 12429-12441.
- Motlagh, N.H., Bagaa, M., and Taleb, T., 2017, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," *IEEE Communications Magazine*, 55(2), pp. 128-134.
- Rao, G.S., Shyam Mohan, B.S., and Sharma, S., 2021, "IoT-Based Smart Air Quality Monitoring System Using LoRa and Machine Learning," *IET Wireless Sensor Systems*, 11(3), pp. 103-116.

- Yi, W.J., Choi, J., and Saniie, J., 2022, "Compressed Sensing Based Air Quality Monitoring Framework With Edge Computing," *IEEE Transactions on Industrial Informatics*, 18(3), pp. 1821-1830.
- Kumar, A., Singh, R., and Rawat, U., 2022, "Edge-Assisted IoT Framework for Real-Time Air Pollution Monitoring," *Journal of Network and Computer Applications*, 197, pp. 103282.
- Heinzelman, W.R., Chandrakasan, A., and Balakrishnan, H., 2000, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of IEEE HICSS*, pp. 1-10.
- Sharma, G., Krishna, A., and Verma, H.K., 2020, "Improved DV-Hop Localization Algorithm for Wireless Sensor Networks," *Wireless Networks*, 26(4), pp. 2535-2549.
- Gui, L., Val, T., Wei, A., and Dalce, R., 2015, "Improvement of Range-Free Localization Technology by a Novel DV-Hop Protocol in Wireless Sensor Networks," *Ad Hoc Networks*, 24(Part B), pp. 55-73.
- Chen, H., Sezaki, K., Deng, P., and So, H.C., 2008, "An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks," *Proceedings of IEEE ICIT*, pp. 1-5.
- Kaur, A., and Bhatt, N., 2021, "Machine Learning Algorithms for Node Localization in Wireless Sensor Networks," *Wireless Personal Communications*, 119(4), pp. 3261-3277.
- Srivastava, S., and Singh, M., 2021, "Prevention and Control of Pollution Using Machine Learning: A Systematic Review," *Journal of Cleaner Production*, 283, pp. 124549.
- Mendez, C.M., Labrador, M.A., and Ramachandran, K., 2013, "Data Interpolation Using Kriging in Wireless Sensor Networks for Air Pollution Monitoring," *Proceedings of IEEE PerCom Workshops*, pp. 333-338.
- Li, X., Shi, H., and Shang, Y., 2020, "A Survey on Localization Algorithms for Wireless Ad Hoc Networks," *IEEE Access*, 8, pp. 106726-106754.
- Hoang, M.T., Zhu, Y., Yuen, B., Rebbeck, T., Tan, U.X., Dutkiewicz, E., and Vesilo, R., 2019, "A Dynamic Compressed Sensing Technique for RF-Based Localization," *IEEE Sensors Journal*, 19(9), pp. 3508-3521.
- Guo, X., Liu, Z., and Ren, Y., 2022, "A Federated Learning Framework for Privacy-Preserving Localization in WSNs," *IEEE Internet of Things Journal*, 9(18), pp. 17583-17594.
- Zhou, Y., Yang, L., and Liu, B., 2020, "Deep Learning-Based Localization Method for Wireless Sensor Networks," *IEEE Access*, 8, pp. 7235-7248.
- Peng, R., and Sichitiu, M.L., 2006, "Angle of Arrival Localization for Wireless Sensor Networks," *Proceedings of IEEE SECON*, pp. 374-382.
- Tarrio, P., Bernardos, A.M., and Casar, J.R., 2011, "Weighted Least Squares Techniques for Improved Received Signal Strength-Based Localization," *Sensors*, 11(9), pp. 8569-8592.
- Khan, M.A., Khan, A., and Rehman, M.H., 2023, "Deep Reinforcement Learning for Intelligent Localization in UAV-Assisted Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, 72(4), pp. 5179-5193.