IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2020, accepted July 9, 2020, date of publication July 14, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009171

A Hybrid Approach Using Multistage Collaborative Calibration for Wireless Sensor Network Localization in 3D Environments

LIANG XU[1], (Member, IEEE), ZHILIANG LI[1], AND XIUXI LI[2]

[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Chemistry and Chemical Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Liang Xu (celiangxu@gdut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 21376091, and in part by the Guangdong Provincial Key Laboratory of Green Chemical Product Technology under Grant GC201812.

ABSTRACT Accurate localization is critical in the internet of things (IOT), especially for wireless sensor networks (WSNs). Location estimation can be affected by factors such as node density, topological diversity, and sensor coverage. As such, we propose a hybrid approach using multistage collaborative calibration for wireless sensor network localization, specifically in 3D environments. This technique integrates a Modified version of Light Gradient Boosting Model (MLGB), which is based on a regression scheme, a cooperative methodology, and a fine calibration model for collaborative fusion. These techniques were combined with quadrilateral shrunk centroid (QSC) and distance vector hop algorithms, using a multi-communication radius and an improved frog-leaping algorithm (DVMFL). In the first step, MLGB was used to correct for inhomogeneous localization estimation errors and RSSI data sparsity. As a result, the model is capable of adapting to high topological diversity (i.e., C-shape, H-shape, S-shape, and O-shape).Successive steps further improved prediction accuracy by using a screening cooperative anchor node strategy to increase node density and enhance the QSC-DVMFL fusion framework for fine position estimation. The proposed methodology was assessed in a series of validation, comparing it to other techniques. The results demonstrated a clear effectiveness and adaptability across a variety of factors that typically affect WSN localization.

INDEX TERMS Wireless sensor network, localization, multistage collaborative calibration technique, modified version of light GBM, quadrilateral shrunk centroid.

I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted increased attention in recent years, due to unprecedented technological progress in multiple engineering disciplines. WSNs are a type of wireless network consisting of hundreds or even thousands of sensors in a self-organizing, multi-hop structure. Information transmission between nodes relies on a wireless network technology, such as, Zigbee, and WIFI.

This architecture is often used to monitor, acquire, process, and transmit perceptual object information within the coverage range. WSNs are a critical component in the internet of things (IOT) and have a variety of military and industrial applications [1]. They have been used for environmental monitoring [2], medical procedures [3], precision agriculture [4], animal tracking [5], remote sensing (during natural disasters) [6], explosives detection [7], and earthquake studies [8]. Although these application environments differ significantly, each requires accurate location information. As such, accurate localization techniques are critical for location-based WSN services. This study focuses specifically on node localization in 3D environments.

A. MOTIVATION

The primary goal of localization is to precisely identify node positions at any given time. Existing localization techniques can be classified as either range-based [9], [10] or range-free [11], [12] models. Range-based algorithms are

The associate editor coordinating the review of this manuscript and approving it for publication was Jagruti Sahoo.

typically more accurate as they estimate the distances between unknown and known (anchor) nodes. Several techniques have been developed to estimate node locations through direct measurements. These include the received signal strength indicator (RSSI), angle of arrival (AOA), time of arrival (TOA), and time difference of arrival (TDOA) algorithms. RSSI data are available in most of wireless network technologies, such as, Zigbee, and are often preferred for identifying radio connectivity information among neighboring nodes. As such, it is often used to estimate approximate locations for unknown nodes using proximity information (e.g., coordinates) based on a dynamic signal propagation model which is introduced in Section III, without additional hardware.

Machine learning (ML) has recently been applied to position estimation for sensor nodes [13], [14]. WSN localization techniques utilize various ML models, such as k-means or fuzzy c-means [15], random forests [16], artificial neural networks (ANN) [17], fuzzy logic (FL) [18], support vector machines (SVMs) [19], Bayesian models [20], principal component analysis (PCA) [21], and semi-supervised learning [22]. Most ML-based techniques consider localization to be a multi-classification problem. However, position estimation for target nodes requires exact coordinate values.

ML-based localization cannot eliminate absolute position estimation errors. In addition, accumulative estimation errors can increase along the edges of a cell. For example, FIGURE 1 shows an unknown node that appears to be positioned at the centroid o in the cell. However, it is actually located at point c. The accumulative error in this case is $\sqrt{2}\big/_{2}D/M$, where D represents the size of the network coverage range and M is the number of categories. Positioning errors for the majority of algorithms primarily focus on marginal areas in the range of sensing coverage, as shown in FIGURE 2. Known (anchor) nodes are often distributed non-uniformly, causing positioning errors to be lower in the center and higher along the edges. This type of irregular network topology is the most significant factor affecting estimation precision (see FIGURE 3). RSSI-based positioning algorithms suffer from uncertainty and nonlinearity, induced by multipath attenuation or non-line-of-sight propagation in the surrounding environment. As such, RSSI data are typically sparse and can include missing values near the edges of a coverage range. However, conventional machine learning methods are sensitive to sparse data, which can be problematic in remote sensing applications. Therefore, a new machine learning technique is required, to address the problems caused by non-uniform estimation errors and irregular network topologies in WSN localization.

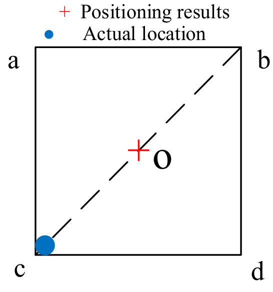


FIGURE 1. Maximum positioning errors for the classification algorithm.

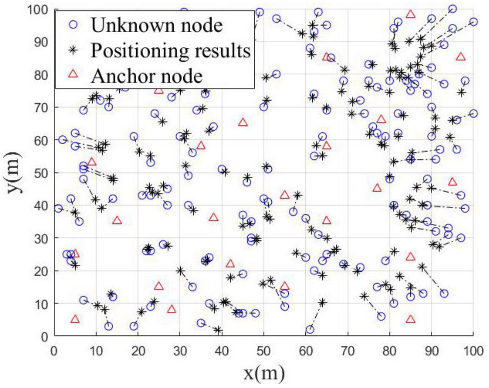
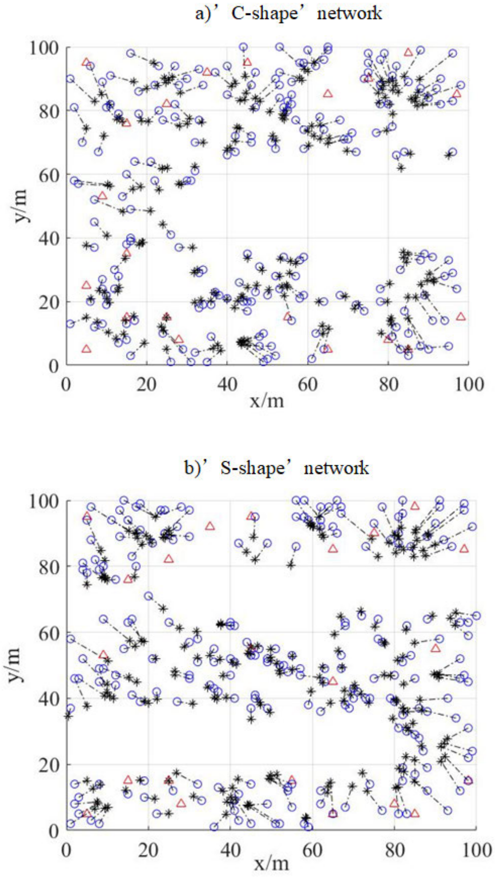


FIGURE 2. Sample positioning errors, which are more concentrated in marginal areas.


a)’ C-shape’ network

b)’ S-shape’ network

FIGURE 3. Positioning results for different network topologies (e.g., C-shape and S-shape).

B. CONTRIBUTIONS

This study investigates key factors affecting WSN location precision in 3D space, including non-uniform errors in position estimation, irregular network topologies, and RSSI data sparsity.

A novel hybrid localization technique is proposed using multistage collaborative calibration to overcome inhomogeneous estimation errors and adapt to sparse or topologically diverse networks (i.e., C-shape, H-shape, S-shape, and O-shape). The contributions of this work can be summarized as follows:

- A hybrid localization technique is presented, using multistage collaborative calibration, for wireless sensor networks in 3D environments.
- Rough localization is performed using a modified light gradient boosting model (MLGB) with a novel loss function, designed to address non-uniform position estimation errors, RSSI data sparsity, and irregular network topology.
- A new collaborative node screening strategy is developed and a selection preference is introduced to reduce positioning errors by increasing cooperative anchor nodes.
- A new cooperative fusion calibration framework is proposed, which combines quadrilateral shrunk centroid and distance vector hop algorithms with a multi-communication radius and an improved frog-leaping algorithm, to estimate the fine positions of target nodes.

The remainder of this paper is organized as follows. Section II introduces related work on WSN localization. Section III provides a detailed description of our hybrid methodology. Experimental results and a detailed performance analysis are included in Section IV. Section V concludes the paper.

II. THE RELATED WORK

Localization is a critical component in a variety of WSN applications. Conventional techniques often focus on estimation optimization [23], [24]. However, machine-learning models have recently been included to improve localization accuracy [25]. These integrated techniques include k-means clustering [26], artificial neural networks (ANNs) [17], [27], fuzzy logic (FL) [18], [28], [29], support vector machines (SVMs) [15], [19], Bayesian optimization [20], principle component analysis (PCA) [21], and semi-supervised [30] or deep learning [31].

Bernas and Płaczek [26] improved localization accuracy using k-means and fuzzy c-means techniques. Banihashemian *et al.* [27] introduced a PSO-based neural network for WSN calibration. This approach utilized hop counts to reduce error rates during localization. Gharghan *et al.* [17] used an artificial neural network (ANN) to locate mobile sensor nodes and improve localization performance with a PSO. This algorithm produced higher accuracy than comparable methods.

Yun *et al.* [28] proposed a fuzzy logic (FL) system to derive additional weights for centroid-based algorithms, using a resultant force vector to adjust locations during the fuzzy process. Phoemphon *et al.* [18] later proposed a fuzzy centroid method to adjust centroid estimation precision. This approach utilized the Takagi-Sugeno-Kang model, in which a weighted average was used to determine the output given five predefined rules. Fuzzy localization algorithms, based on centroid algorithms with fuzzy Mamdani and Sugeno inference systems, have been shown to increase accuracy by using flow measurements through a wireless channel [29].

Wang *et al.* [19] developed a range-free localization algorithm using SVM classification. In this process, the WSN region was divided into multiple grids, with each node assigned to one of the grids by an SVM. A fast SVM-based localization algorithm was also reported by Zhu and Wei [15]. The algorithm used similarities to divide feature spaces into support vector groups. In a study by Guo *et al.* [20], localization techniques were based on an iterative variational Bayesian interface, which was used to reduce off-grid faults caused by inaccurate approximations. Li *et al.* [21] introduced a non-convex robust PCA algorithm to eliminate outliers. Kumar *et al.* [30] presented a localization technique that used a semi-supervised hidden Markov model (HMM) for mobile WSN nodes. The algorithm worked well in both indoor and outdoor environments, while requiring fewer training data than some models. Chang [31] proposed a new deep learning-based localization technique for high-speed mobile objects such as autonomous vehicles. The algorithm included rough localization, which employs a modified Kalman filter to produce rough location estimates and fine localization, based on a deep learning technique.

These machine learning-based localization techniques classify unknown nodes into an appropriate grid to estimate node positions in a network. As such, these techniques depend on the classification performance of machine learning models. However, WSN localization is not solely a classification problem and location estimation using machine learning has inherent limitations.

The majority of localization techniques depend on known (anchor) nodes to estimate unknown node positions. However, unknown nodes can also be used to improve precision in a process called cooperative localization [32], [33]. In this process, approximate node positions are iteratively updated using maximum likelihood estimation [33]. Hu *et al.* [34] proposed a new cooperative positioning algorithm, applicable to the internet of things, which included factor graphing based on Fisher information matrix theory. However, the key problem in cooperative localization is choosing the most appropriate unknown nodes for a given cooperative (known) node.

Several recent studies have investigated the location of unknown WSN nodes in 3D environments. In [35], two models: hybrid particle swarm optimization (HPSO) and biogeography optimization for range-based models. These algorithms consider transmission range and anisotropic WSN properties in 3D environments. Kumar *et al.* [36] extended this model into a range-free HPSO for 3D node localization. However, the algorithm does not consider computational cost or runtime, which reduces the network life expectancy. To address this problem, Raguraman *et al.* [37] proposed two dimensionality-based 3D node localization algorithms using the particle swarm optimization framework. Both algorithms exhibited a lower computational cost and minimized the error rate in location estimations.

The limitations of single methods for improving localization precision have popularized hybrid localization approaches, integrating a variety of models. Phoemphon *et al.* [18] proposed a hybrid technique using fuzzy logic and an extreme learning machine, with vector particle swarm optimization for WSN localization. A similar model was developed by Zhu and Wei [38], using SVM and fuzzy *c*-means. The algorithm provided a tradeoff between localization accuracy and training time. Phoemphon *et al.* [39] further improved node localization in areas with obstructions. This integrated technique, which combined node segmentation with improved particle swarm optimization (NS-IPSO), increased the accuracy of estimated distances.

This study proposes a new hybrid technique for WSN localization, specifically in 3D environments. This multi-stage collaborative calibration strategy was used to iteratively enhance localization accuracy. First, an MLGB model was used for a rough estimation of node positions. A cooperative localization technique was then used to improve anchor node density. A fusion calibration framework was further applied for fine localization.

III. THE HYBRID APPROACH

FIGURE 4 provides an overview of the proposed hybrid technique, which includes three primary components:

- *Rough localization (see Section A)* – A rough localization of target nodes was calculated using a modified version of the light gradient boosting model (MLGB), in which anew loss function was developed for WSN localization. After acquiring RSSI information, a rough localization estimation for x, y, and z is performed using an MLGB.
- *Screening collaborative nodes with selected preference accuracy (see Section B)* – This stage includes three steps. First, nodes with high positioning accuracy were selected from a set of unknown nodes based on RSSI thresholds. Second, unknown nodes that were less affected by the environment were chosen as candidate collaboration nodes, using a subset judgment strategy. Third, high-precision nodes were selected as collaboration nodes using an anchor node replacement strategy.
- *Cooperative fusion calibration framework (see Section C)* – This stage utilizes quadrilateral shrunk centroid and distance vector hop algorithms, based on a multi-communication radius and an improved frog leaping algorithm, for localization estimation of target nodes.

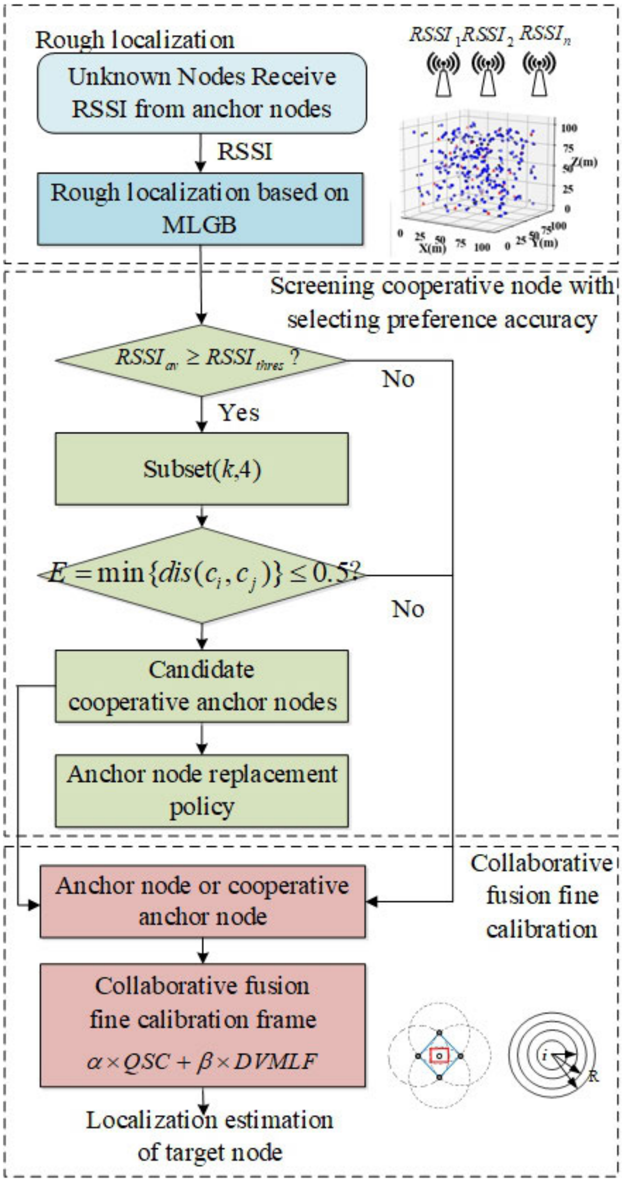

Rough localization
$RSSI_1$ $RSSI_2$ $RSSI_n$
Unknown Nodes Receive RSSI from anchor nodes
RSSI
Rough localization based on MLGB
Z(m)
X(m)
Y(m)
Screening cooperative node with selecting preference accuracy
$RSSI_{av} \geq RSSI_{thres}$?
No
Yes
Subset(k,4)
$E = \min\{dis(c_i, c_j)\} \leq 0.5$?
No
Candidate cooperative anchor nodes
Anchor node replacement policy
Collaborative fusion fine calibration
Anchor node or cooperative anchor node
Collaborative fusion fine calibration frame
$\alpha \times QSC + \beta \times DVMLF$
r
R
Localization estimation of target node

FIGURE 4. A flowchart for the proposed hybrid approach.

A. ROUGH LOCALIZATION BASED ON MLGB

Unlike conventional ML-based localization algorithms, based on a classification model, rough localization estimates approximate positions using regression MLGB. The purpose of this stage is to eliminate inhomogeneous errors and resolve RSSI data sparsity, as well as predict positions for unknown nodes (by considering different network topologies in 3D).

The MLGB model is based on the light gradient boosting model (Light GBM) [40], which is an improved version of a gradient-boosting decision tree (GBDT) [41]. Light GBM is a widely used machine-learning algorithm featuring a decision tree as its base classifier, which addresses limitations in large-scale sparse datasets. Two new models, gradient-based one-sided sampling (GOSS) and exclusive feature bundling (EFB) [40], are proposed to improve the light GBM performance by reducing runtime and increasing accuracy.

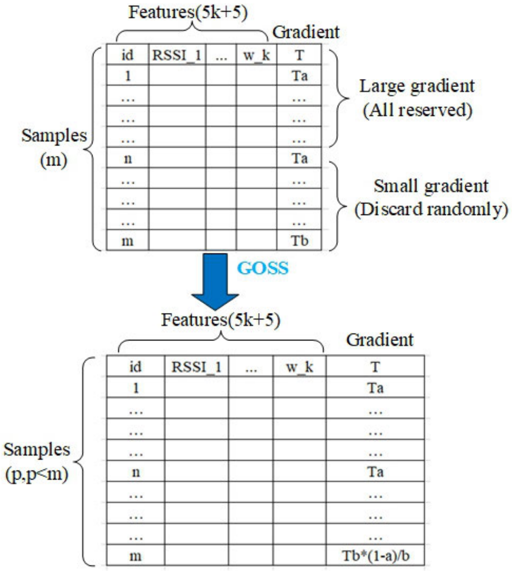
1) LIGHT GBM

This section provides a detailed overview of two new techniques (GOSS and EFB) adopted in the light GBM model.

(1) GOSS was used to pre-process sample data during the training process. The negative gradient of the loss function for each sample i can be expressed as:

$$T_i = -\frac{\partial L(y_i, f_{t-1}(x_i))}{\partial f_{t-1}(x_i)}, \quad \mathrm{i} = 1, \ldots m, \tag{1}$$

where L is the loss function, y is the value of each sample, and f is the boosting tree function. The working theory for GOSS is that all samples with large gradient values are preserved, but only samples with small gradient values are randomly sampled. In FIGURE 5, T_a and T_b are respectively the large and small gradient values. After sampling, n_a large gradients and n_b small gradients were combined to form a new training set. The sampled data were then amended to preserve the data distribution. In other words, small gradient values were multiplied by a weighting coefficient $\frac{1-a}{b}$, where a and b represent sampling ratios for large and small gradient data, respectively.



id	RSSI_1	...	w_k	T
1				Ta
...				...
...				...
...				...
n				Ta
...				...
...				...
...				...
m				Tb

id	RSSI_1	...	w_k	T
1				Ta
...				...
...				...
...				...
n				Ta
...				...
...				...
...				...
m				Tb*(1-a)/b

FIGURE 5. Gradient-based one-sided sampling (GOSS).

(2) The exclusive feature bundling (EFB) technique was applied to sparse data features. Ten RSSI samples, in which 0 or null is the missed value, are shown in FIGURE 6. The GOSS algorithm was applied to the data in the n^{th} iteration. The features RSSI_1 (sparse data) and RSSI_3 in Table A are mutually exclusive, meaning they never take nonzero values simultaneously. The problem of data sparsity can be solved by bundling two features into one. In this process, RSSI_3 must be distinguishable from the new bundling feature (RSSI_1&3), as shown in Table B. This can be achieved by adding an offset value (for example, 95) to RSSI_3. The EFB algorithm can then be used to compensate for RSSI data sparsity by bundling exclusive features into fewer dense features.

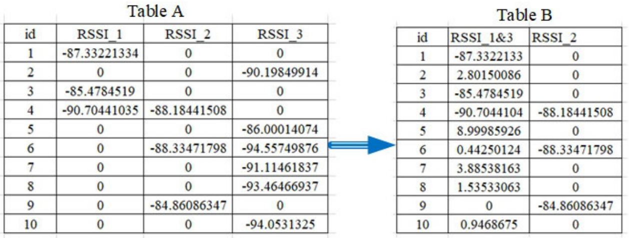


Table A

id	RSSI_1	RSSI_2	RSSI_3
1	-87.33221334	0	0
2	0	0	-90.19849914
3	-85.4784519	0	0
4	-90.70441035	-88.18441508	0
5	0	0	-86.00014074
6	0	-88.33471798	-94.55749876
7	0	0	-91.11461837
8	0	0	-93.46466937
9	0	-84.86086347	0
10	0	0	-94.0531325

Table B

id	RSSI_1&3	RSSI_2
1	-87.3322133	0
2	2.80150086	0
3	-85.4784519	0
4	-90.7044104	-88.18441508
5	8.99985926	0
6	0.44250124	-88.33471798
7	3.88538163	0
8	1.53533063	0
9	0	-84.86086347
10	0.9468675	0

FIGURE 6. Exclusive feature bundling (EFB).

Divided node gains can then be calculated in the decision tree using:

$$(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right), \tag{2}$$

where $A_l = \{x_i \in A : x_{ij} \le d, A_r = \{x_i \in A : x_{ij} > d, B_l = \{x_i \in B : x_{ij} \le d, A_r = \{x_i \in B : x_{ij} > d$.

During node splitting, the leaf node with the largest divided gain is selected for growth. This process is a leaf-wise growth strategy, designed to produce the smallest error. Trees were grown with deep constraints in order to decrease the complexity of the model.

2) A LOCALIZATION ALGORITHM BASED ON MLGB

RSSI data are often sparse and noisy. As such, they exhibit both uncertain and nonlinear characteristics and may contain missing values, due to the non-uniform deployment of WSNs. As such, EFB and histogram equalization algorithms were used to overcome sparsity and noise, respectively. In the training process, GOSS was used for data pre-processing to achieve a tradeoff between the number of samples and the accuracy of the learning decision tree.

The loss function for the light GBM regression task can be expressed as:

$$L^{(t)} = \sum_{i=1}^{n} (y_i - (f_{t-1}(x_i) + h_t(x_i)))^2 \tag{3}$$

where h represents a decision tree function.

The first stage, rough localization, implements an approximate position estimation. A loss function was used to measure the difference between an estimated result and the actual position. Positioning error in WSNs are closely related to the communication radius and location accuracy will vary greatly with different radii. Thus, a set communication radius was introduced into the model for localization estimation. The number of leaf nodes (T) and the 2-norm of the parameter φ_T were concurrently added to the function to control the complexity of the model and prevent model overfitting. This new model is a Modified Light GBM (MLGB) algorithm.

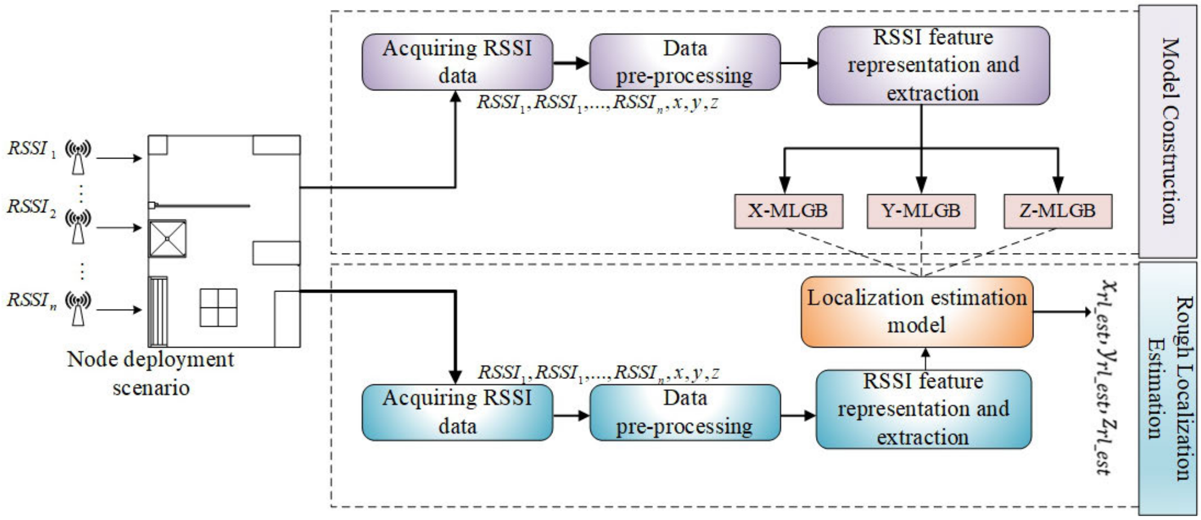

Acquiring RSSI data
Data pre-processing
RSSI feature representation and extraction
RSSI1, RSSI1,..., RSSIn, x, y, z
Model Construction
X-MLGB
Y-MLGB
Z-MLGB
RSSI1
RSSI2
RSSIn
Node deployment scenario
Localization estimation model
x_rl_est, y_rl_est, z_rl_est
RSSI1, RSSI1,..., RSSIn, x, y, z
Acquiring RSSI data
Data pre-processing
RSSI feature representation and extraction
Rough Localization Estimation

FIGURE 7. **A flowchart for the rough localization algorithm.**

The new loss function can then be written as:

$$L^{(t)} = \frac{\sum_{i=1}^{n} (y_i - (f_{t-1}(x_i) + \nu \cdot h_t(x_i)))^2}{n \cdot R} + \gamma T + \frac{\lambda}{2} \|\varphi_T\|_2^2, \tag{4}$$

where R is the node signal radius and ν is the learning velocity ($0 < \nu \leq 1$). The terms γ and λ are regularization parameters, T is the number of leaf nodes, φ_T is a decision tree parameter vector, $f_{t-1}(x_i)$ is a strong classifier acquired from the previous iteration, and $h_t(x_i)$ is a weak classifier in the t^{th} iteration. The MLGB algorithm can compensate for inhomogeneous localization estimation errors and RSSI data sparsity. In addition, regression allows it to adapt to different network topologies.

A flowchart of these five rough localization steps is shown in FIGURE 7. The detailed procedure can be described as follows.

Step 1: Acquiring RSSI data

Assuming anchor nodes $S_i (i = 1, \ldots, k)$ were deployed in the WSN coverage range. RSSI values were collected from every unknown node and anchor node S_i. The dataset $[RSSI_1, \ldots, RSSI_k, x_j, y_j, z_j]$ was then established by recoding the coordinates (x_j, y_j, z_j) of every node and the RSSI values corresponding to sets of unknown and anchor nodes. In this notation, the subscript j denotes the j^{th} node ($j = 1, \ldots, m$), where m is the total number of training samples.

Step 2: Data pre-processing

RSSI errors caused by noise interference were reduced by collecting multiple groups of data for each unknown node. The data were then processed using a Gaussian filter model. The mean value (μ) and variance (σ) of the RSSI dataset were calculated using:

$$\begin{cases} \mu = \frac{1}{n} \sum_{i=1}^{n} RSSI_i \\ \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (RSSI_i - \mu)^2. \end{cases} \tag{5}$$

Abnormal data will inevitably be included as RSSI signals suffer from environmental interference. As such, a local outlier detection algorithm was used to eliminate these abnormal data.

The local outlier factor (LOF) can be expressed as:

$$LOF_k(q) = \frac{1}{|N_k(q)|} \sum_{p \in N_k(q)} \frac{lrd_k(p)}{lrd_k(q)} \tag{6}$$

where $N_k(q)$ is the k^{th} neighborhood of sample q and $lrd_k(p)$ and $lrd_k(q)$ respectively represent the local reachable densities for samples p and q, using the distance between each as a measurement [42]. These can be expressed as:

$$lrd_k(q) = 1 \Bigg/ \frac{\sum_{p \in N_k(q)} reach\_dist_k(p, q)}{|N_k(q)|},$$

$$lrd_k(p) = 1 \Bigg/ \frac{\sum_{p \in N_k(p)} reach\_dist_k(p, q)}{|N_k(p)|}.$$

An LOF value greater than 1 implies a higher outlier degree for a given sample, meaning it is more likely abnormal than normal. Such data were excluded from the experiments.

Step 3: RSSI feature representation and extraction

Model performance depends on the RSSI feature representation. In this step, the relevant 5-dimensional statistical features (i.e., variance, maximum, minimum, mean, and range) were extracted from the RSSI set to reflect trends and data volatility. RSSI values were then converted into distances using a dynamic signal propagation model.

The common signal propagation model is given as:

$$P(d) = P(d_0) - 10e_p lg(\frac{d}{d_0}) + X_\sigma \tag{7}$$

where $P(d)$ is the received signal strength at point d from a node, e_p is the path loss exponent representing the rate at which RSSI decays with distance and X is the Gaussian random noise. Thus, according to (7), the signal propagation distance can be calculated based on the RSSI.

Given three adjacent nodes (A, B, and C) corresponding to an unknown node n, the RSSI value between A and B or A and C can respectively be represented as:

$$\begin{cases} P(d_B) = P(d_0) - 10e_p lg(\dfrac{d_{AB}}{d_0}) + X_\sigma \\ P(d_C) = P(d_0) - 10e_p lg(\dfrac{d_{AC}}{d_0}) + X_\sigma, \end{cases} \tag{8}$$

Accurate values of e_p can reduce distance estimation errors. If the positions of nodes A, B, and C are known with sufficient precision, d_{AB} and d_{AC} can be calculated to the required accuracy. Equation (8) can then be solved for e_p value. Distance errors can be amended by calculating e_p dynamically. Thus, such model is called as the dynamic signal propagation model.

Shorter distances inherently produce less environmental interference and signal attenuation. Thus, position estimation errors will be smaller for unknown nodes that are closer to anchor nodes. RSSI values are also given higher weights between both nodes. The modified weighting function for each known node can be expressed as:

$$\omega_i = \frac{d_{max} - d_i}{\sum_{i \neq max} d_{max} - d_i}, \tag{9}$$

where d_i represents the distance between an unknown node and the i^{th} anchor node. The term d_{max} denotes the maximum value among all distances between both node sets. Two new combined features, composed of several first-order discrete features, were introduced to solve nonlinear problems. The signal-distance cross feature can then be expressed as $R_{dj} = RSSI_j \times d_j$ and the weighted RSSI feature is given by $R_{\omega j} = RSSI_j \times \omega_j$. Here, R_{dj} and R_{wj} are j-dimensional ($j = 1 \ldots k$) signal distance cross features and weighted RSSI features for unknown nodes, respectively.

Step 4: off-line Model training

The feature data consisted of (5k + 5)×m rows, including k-dimensional RSSI data, distances, weights, signal distances, and weighted RSSI features, as well as 5-dimensional statistical parameters. The feature data and the coordinates (X, Y, and Z) for any node could be combined to form three training datasets: data$_x$, data$_y$, and data$_z$, respectively. After data pre-processing, the position estimation models in each of the three coordinate directions (X, Y, and Z) were respectively labeled X-MLGB, Y-MLGB, and Z-MLGB.The models were then trained using the three datasets.

Step 5: Position estimation

After target node data were acquired and pre-processed, RSSI feature extraction was conducted using data with a structure similar to that of the training samples. These data were input to the X-MLGB, Y-MLGB, and Z-MLGB models, which provided unknown nodes with rough localization estimates $(x_{rl\_est}, y_{rl\_est}, z_{rl\_est})$. This approach can be represented systematically as shown in the algorithm 1.

MLGB-based rough localization techniques can overcome uniform error distributions in position estimates and sparsity in RSSI data. They are also highly robust and can adapt to different network topologies. However, experimental results have shown that estimation errors can be as high as 10%. Therefore, other techniques are required to further improve localization precision.

Algorithm 1 Rough Localization Based on MLGB

```
Input: Known $(x_{AN}, y_{AN}, z_{AN})$ and unknown nodes $(i \in 1, 2, 3, \ldots\ldots, n)$.
for i=1 to n do:
  $[RSSI_1, \ldots, RSSI_k]$ ← Collect signal strength values
  for $j = 1$ to m do:
  $\mu, \sigma^2$ ← Apply a Gaussian filter
  end for
  $LOF_k(q) = (\sum_{p \in N_k(p)} lrd_k(p)/lrd_k(q))/|N_k(q)|)$
  $[RSSI_1, \ldots, d_1, \ldots, \omega_k]$ ← Construction feature engineering
  X-MLGB, Y- MLGB, Z- MLGB ← $[RSSI_1, \ldots, d_1, \ldots, \omega_k, x, y, z]$
end for
Output: Estimated target node position $(x_{rl\_est}, y_{rl\_est}, z_{rl\_est})$
```

B. SCREENING COOPERATIVE NODES WITH SELECTED PREFERENCE ACCURACY

Anchor node density is a key factor influencing location estimation performance. In this stage, a cooperative localization strategy is introduced to improve node distributions, by adding a cooperative anchor node. Unknown WSN nodes can not only communicate with anchor nodes, but also with each other. In this way, anchor nodes can take advantage of cooperation between unknown nodes, producing cooperative anchor (CA) nodes that can be used to collaboratively estimate unknown node positions.

The accurate selection of CA nodes is critical in this process, as mistaking unknown nodes for CA nodes can decrease accuracy and increase algorithm complexity [33], [43], [44]. Cumulative errors from CA node selection can be reduced by ensuring that unknown nodes are identified with sufficiently high precision. In this study, a new strategy is proposed for screening CA nodes with preference accuracy. The method is composed of three steps: an RSSI threshold-based filtering node, a subset judgement policy, and an anchor node replacement strategy.

1) RSSI THRESHOLD-BASED FILTERING NODES

RSSI values are easily affected by external conditions. In this stage of the proposed algorithm, RSSI thresholds were determined empirically in various scenarios. Unknown nodes with high precision were then screened out as candidate CA nodes using the resulting thresholds. These experiments demonstrated that larger RSSI values produced estimated values that were closer to the actual values. This is in agreement with the signal propagation model, which predicts that smaller RSSI values yield greater errors between measured and actual data (see FIGURE 8). Empirical RSSI thresholds are provided in Table 1, where the included parameters are $d_0 = 1\text{m}$, $P(d_0) = -40\text{dBm}$, and X_σ is a normal distribution with a mean of zero and a variance of 2.

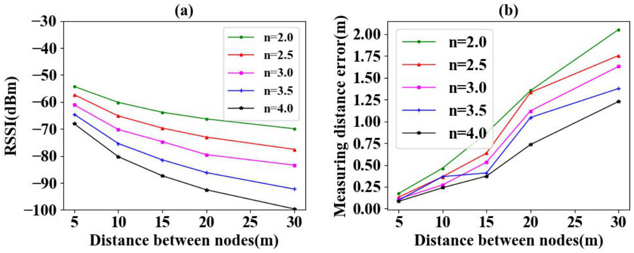


FIGURE 8. The relationship between measurement distance errors and signal strength.

TABLE 1. Relationship between Path loss exponent and RSSI threshold.

Path loss exponent	threshold value for RSSI
2~2.5	-65dBm
2.5~3	-80dBm
3~3.5	-90dBm
3.5~4	-100dBm

An unknown node may include k RSSI values at the same position for sampling time k. As such, a Gaussian model was used to process these data and reduce the impact of interference noise. The term $RSSI_{av}$ refers to the average of RSSI values, which can be expressed as:

$$RSSI_{av} = \frac{1}{n}\sum\nolimits_{i=1}^{n} RSSI_i, \tag{10}$$

where n is the number of anchor nodes and $RSSI_i$ is the RSSI value between the unknown node and the i^{th} anchor node. The corresponding decision rule is then given by:

$$RSSI_{av} \geq RSSI_{thres}, \tag{11}$$

where $RSSI_{thres}$ can be determined from the values in Table 1. Any node that satisfies the conditions described in these equations is considered high precision. In this stage, unknown nodes with low accuracy can be omitted and only a few nodes with high precision need to be retained as candidate cooperative nodes.

2) SUBSET DETERMINATION POLICY

A subset judgment policy is proposed to further eliminate environmental impacts on RSSI data. Experimental results indicated that larger RSSI values resulted in smaller distance errors and higher positioning accuracy. As such, the k nodes with the largest RSSI values were selected from the nearest neighbors of a given unknown node. Four nodes from each set were then selected and the QSC algorithm (a new centroid localization method, see section C.1)) was used to calculate estimated positions for unknown nodes. Typically, k positions for each unknown node approach the same position. Subsets were constructed in which any four of k nodes with the largest RSSI values composed a set $(k, 4)$. These four nodes could then be used to estimate unknown node positions with the QSC algorithm, producing k estimated positions. The minimum error among these estimations is given by:

$$E = min\{dis(c_i, c_j),\ \forall i \in [1, k],\ j \in [1, k]\ and\ j \neq i\}, \tag{12}$$

where i and j are the number of subsets, c_i is a position estimation based on the i^{th} subset, and $dis(c_i, c_j)$ indicates the distance error between the i^{th} and j^{th} position estimates. The Euclidean distance was then calculated for each set. The corresponding judgement condition is given by:

$$E_s \leq \mu, \tag{13}$$

where μ is a threshold value. This term is a trade-off between the almost non-existent nodes with higher localization accuracies (corresponding to smaller values of μ) and nodes with lower localization accuracies (corresponding to larger values of μ). The estimation error was calculated for N unknown nodes in the subset E, resulting in a value for μ that can be expressed as:

$$\mu = \frac{1}{N}\sum\nolimits_{r=1}^{N} E_r; \quad r, s \in [1, N] \tag{14}$$

where E_s is the minimum error in the subset for the s^{th} unknown node and E_r is the minimum error in the subset for the r^{th} unknown node.

Satisfying this condition indicates that a position estimate is accurate, exhibiting limited environmental interference. As such, all unknown nodes that satisfy (12) and (13) were assumed to be candidate CA nodes. This approach was used to eliminate unknown nodes affected by the environment, producing screened nodes with higher precision.

3) ANCHOR NODE REPLACEMENT STRATEGY

The proposed technique increased the number of anchor nodes and improved the precision of position estimates. An anchor node replacement strategy was developed to process candidate

CA nodes and retain those with the highest accuracy are retained. This approach can be described as follows. The position estimate for an unknown node N is represented by the coordinates (x_N, y_N, z_N), which is assumed to be a transforming anchor node. The positions of four anchor nodes can then be expressed as $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$, $C(x_C, y_C, z_C)$, and $D(x_D, y_D, z_D)$. If, for example, node B is an unknown node, the QSC algorithm can be used to process nodes A, C, D, and N. The result is a localization estimate for node B, represented by $B'(x_{B'}, y_{B'}, z_{B'})$. The error between the estimated and actual value is then given by:

$$\sqrt{(x_B - x_{B'})^2 + (y_B - y_{B'})^2 + (z_B - z_{B'})^2} \leq \delta, \tag{15}$$

where δ controls the estimation error ($\delta = 0.2$ m in this study). Smaller values of δ result in smaller errors. If the error for node B satisfies equation (15), the precision of node N is higher. As such, node N is considered the CA node that replaces node B. All unknown nodes can be screened and considered CA nodes using this strategy.

C. COLLABORATIVE FUSION FINE CALIBRATION

In this stage, a collaborative fusion fine calibration framework is proposed, which consists of quadrilateral shrunk centroid (QSC) and distance vector algorithms based on a multi-communication radius and improved frog leaping (DVMFL) models, to produce rough location estimates. The step-by-step procedure for both QSC and DVMFL are combined in the flowchart shown in FIGURE 9.

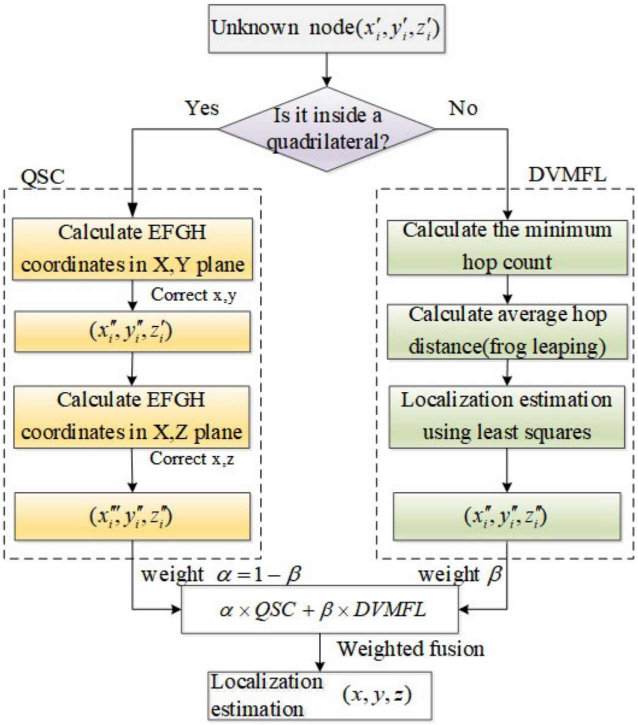

Unknown node(x_i', y_i', z_i')
Yes
Is it inside a quadrilateral?
No
QSC
DVMFL
Calculate EFGH coordinates in X,Y plane
Correct x,y
(x_i'', y_i'', z_i')
Calculate EFGH coordinates in X,Z plane
Correct x,z
(x_i''', y_i'', z_i'')
Calculate the minimum hop count
Calculate average hop distance(frog leaping)
Localization estimation using least squares
(x_i'', y_i'', z_i'')
weight $\alpha = 1 - \beta$
weight β
$\alpha \times QSC + \beta \times DVMFL$
Weighted fusion
Localization estimation
(x, y, z)

FIGURE 9. The collaborative fusion calibration framework.

1) THE QUADRILATERAL SHRUNK CENTROID ALGORITHM

The conventional centroid algorithm is based on a triangle location principal [45]. However, three circles do not necessarily intersect at a point, but form a 2D spatial region that results in position estimation errors. Theoretically, including more nodes in the localization process increases positioning accuracy. As such, we propose a quadrilateral contraction centroid algorithm, with the intention of reducing the area to iteratively approximate the spatial centroid. In this process, the four nodes consist of known (anchor) or unknown (CA) nodes. FIGURE 10 demonstrates a quadrilateral centroid approximating the actual position by shrinking by different quadrilaterals. This 3D algorithm is based on a two-dimensional plane and initially uses the XY plane to make corrections along the X- and Y-axis. It was then applied in the X-Z plane to make corrections along the X- and Z-axes.

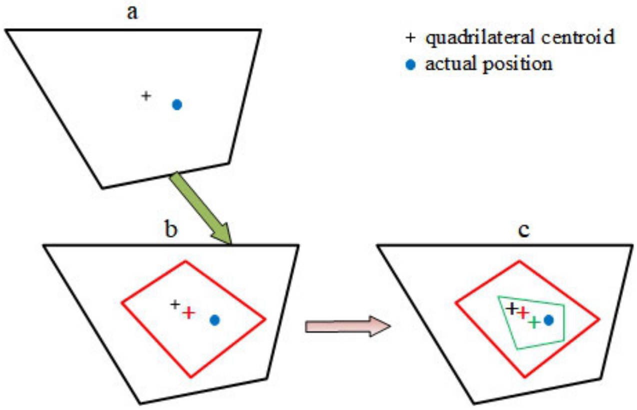


FIGURE 10. The proposed center of mass approximation technique (region shrinking). a) there is an error between the estimated position (quadrilateral centroid) and the actual position; b,c) represents two quadrilateral centroids(red '+', green '+') are gradually approaching to the actual position.

The QSC algorithm utilizes the intersection of four circles to form a quadrilateral region around the target node N and assumes the centroid of this quadrilateral to be the estimated node position. FIGURE 11 shows an example of four nodes that could be either anchor nodes or CA nodes, labeled A, B, C, and D. The distance between any two nodes is considered the radius of a circle. The quadrilateral region is composed of the intersection of two adjacent circles. The centroids of the new quadrilaterals E, F, G, and H, which are formed by the inner intersection of any two adjacent circles, were used as position estimates for node N. This was done by iteratively shrinking the region, as shown in FIGURE 11.

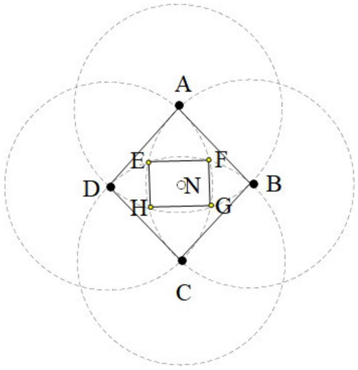


FIGURE 11. An illustration of the quadrilateral centroid shrinking algorithm.

The four intersection points can be represented as The four intersection points can be represented as $E(x_E, y_E)$, $F(x_F, y_F)$, $G(x_G, y_G)$, and $H(x_H, y_H)$. The position of node N can then be determined from:

$$\begin{cases} x = \dfrac{x_E + x_F + x_G + x_H}{4} \\ y = \dfrac{y_E + y_F + y_G + y_H}{4}. \end{cases} \tag{16}$$

Coordinates for E, F, G, and H are based on a two-dimensional rotation matrix, as shown in FIGURE 12. The corresponding transformation can be expressed as:

$$\begin{cases} x_A - x_D = R\cos\theta^* \\ y_A - y_D = R\sin\theta^* \end{cases} \tag{17}$$

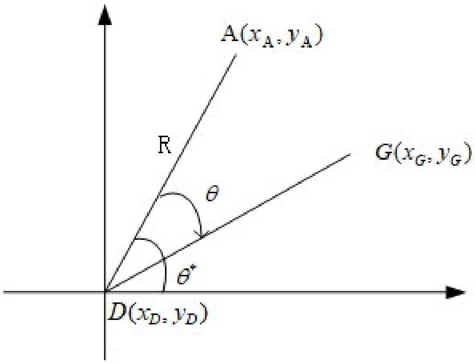


FIGURE 12. **A rotation matrix map.**

where θ^* is the angle between and the x-axis. The vector is formed by rotating the vector through an angle θ, which yields:

$$\begin{cases} x_G - x_D = R\cos(\theta^* - \theta) \\ y_G - y_D = R\sin(\theta^* - \theta). \end{cases} \tag{18}$$

The coordinates for point G can then be calculated as:

$$\begin{bmatrix} x_G \\ y_G \end{bmatrix} = \begin{bmatrix} x_D \\ y_D \end{bmatrix} + \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_A - x_D \\ y_A - y_D \end{bmatrix}. \tag{19}$$

Therefore, the coordinates for G can be calculated from the positions of nodes A and D. Similarly, the coordinates for H, E, and F were determined from the adjacent pairs AB, BC, and CD, respectively. As such, it is critical to determine whether anchor nodes are adjacent to each other. Assuming the order of four anchor nodes is not known, two nodes (B and C) can be randomly selected and a linear equation $F(x) = mx + b$ can be determined. The coordinates (x_A, y_A) and (x_D, y_D) of anchor nodes A and D were used in conjunction with (20) to classify nodes as being on either the same side or opposite sides. This can be represented as:

$$\begin{cases} F(x_A) \times F(y_A) > 0 \quad (\textit{same side}) \\ F(x_D) \times F(y_D) < 0 \quad (\textit{opposite side}). \end{cases} \tag{20}$$

In this process, two distinct cases must be considered: 1) the two nodes (B and D) are on opposite sides and 2) two nodes and C form the edge and diagonal segments of a quadrilateral.

FIGURE 13(a) shows a configuration in which the straight line between nodes A and C forms the diagonal of a quadrilateral. After the first judgment is made, nodes B and D lie on the two sides of the line AC. Four adjacent anchor nodes can be determined from this configuration. FIGURE 13(b) shows the geometry used to make the twice judgement with the four nodes. For the first judgement, nodes A and D lie on the same side of the line BC, as indicated by (18). In the second judgement, it is evident that nodes A and B lie also the same side of the line CD. Thus, nodes B, C, and D form two sides (i.e. BC and CD) of a quadrilateral. FIGURE 13(c) demonstrates that the first judgment is similar to FIGURE 13(b). Nodes B and D lie on opposite sides of the line AC in the second judgment. Thus, nodes A, B, and C construct the edge and a diagonal of a quadrilateral. Therefore, the four anchor nodes are adjacent to each other and can be determined for three different configurations. The accompanying QSC algorithm is described in Algorithm 2.

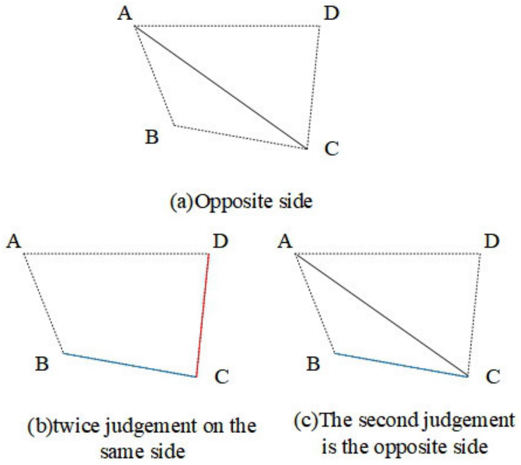

A
D
B
C
(a)Opposite side
A
D
B
C
(b)twice judgement on the same side
A
D
B
C
(c)The second judgement is the opposite side

FIGURE 13. **Three different configurations involving adjacent nodes.**

The primary limitation of this algorithm is that position estimation can be corrected while an unknown node is inside a quadrilateral composed of four anchor nodes. This is demonstrated in FIGURES 14 (a) and (b), where the red triangle represents the anchor node, the green '+' is the estimated position acquired by rough localization, and the black '+' is the QSC algorithm correction result. The blue '.' is the actual position of the target node. However, as shown in FIGURE 14(c), it is possible for an unknown node to lie outside of the quadrilateral. It is also possible for the quadrilateral to be composed of non-anchor nodes as in (FIGURE 14(d)).

Localization errors produced by the QSC algorithm are significantly higher in these cases. As such, the DVMFL algorithm is introduced to overcome these issues.

2) DVMFL ALGORITHM

The DVMFL algorithm is a modified version of the conventional DV-hop method, which is a well-known range-free localization technique. However, DV-hop exhibits two common flaws: 1) hop count and 2) average hop distance errors. As a result, the real distance is often very different while the minimum hop count is mostly the same. However, accurate node positioning depends heavily on the estimation of average hop distance. As such, we propose the

Algorithm 2 Quadrilateral Shrunk Centroid (QSC)

```
Input: Rough location node estimation $(x_{rl\_est}, y_{rl\_est}, z_{rl\_est})$
  $x_{rl\_est}$ as $x_i'$, $y_{rl\_est}$ as $y_i'$, $z_{rl\_est}$ as $z_i'$, i ∈ $1, 2, \ldots, n$,
  $\theta = 60, j = 0$;
for $i = 1$ to $n$ do:
  $j = j + 1 \leftarrow$ Cooperative node
  if $j = 4$ then:
    $E, F, G, H \leftarrow$ Rotation matrices for $x$ and $y$
    $(x_i'', y_i'', z_i') \leftarrow x_i'' = \frac{x_E+x_F+x_G+x_H}{4}, y_i'' = \frac{y_E+y_F+y_G+y_H}{4}$
    $E', F', G', H' \leftarrow$ Rotation matrices for $x$ and $z$
    $(x_i''', y_i'', z_i'') \leftarrow x_i''' = \frac{x_{E'}+x_{F'}+x_{G'}+x_{H'}}{4}, z_i'' = \frac{z_{E'}+z_{F'}+z_{G'}+z_{H'}}{4}$
  else
    continue
  end if
end for
Output: Corrected node $(x_i''', y_i'', z_i'')$
```

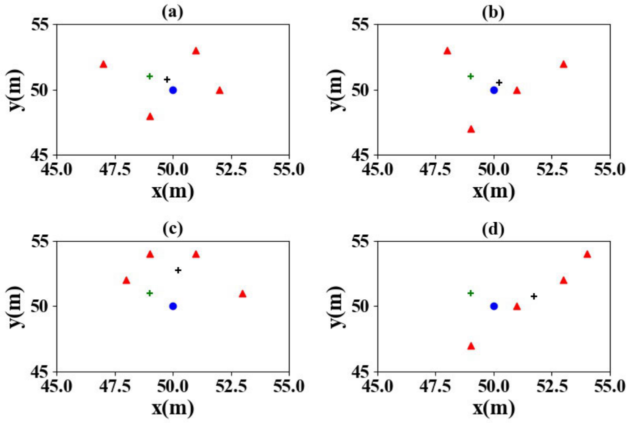


FIGURE 14. Localization correction results for various situations.

DVMFL algorithm to address these two issues. The algorithm is based on a multi-communication radius and an improved frog leaping model, used to optimize the hop distance for anchor nodes and enhance localization performance. This process consists of three steps, which are described in detail below.

Step 1: Minimum hop counts are calculated based on a multi-communication radius

Each anchor node broadcasts two parameters: coordinates and minimum hop count. The initial value of minimum hop count is zero. Each adjacent node then stores the received information and forwards the updated values to neighboring nodes after adding 1 hop value. However, the actual distance between nodes will occasionally differ while the minimum hop count remains the same. To resolve this issue, we propose dividing the signal radius R into DF sections. The minimum hop count from a given node to the anchor node is then DF*R, where *num* is an empirical decimal value ranging from 0 to 1 with a step size of 0.25.

In FIGURE 15, the minimum hop counts for nodes in regions A, B, C, and D (to the anchor node *i*) were 0.25R, 0.5R, 0.75R, and R, respectively. Neighboring nodes then receive and forward anchor node information within the communication radius R. After the end of message transmission, each node obtains the minimum hop count to the anchor node *i*. Refining a hop count between nodes can make the minimum hop number more accurate.

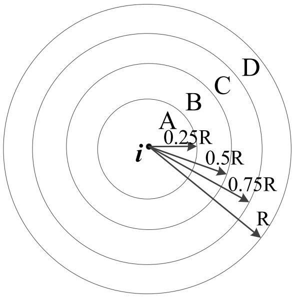


FIGURE 15. An example of a multi-communication radius.

Step2: The average hop-distance is calculated using a modified frog leaping algorithm

In this section, we introduce an improved frog leaping algorithm to optimize the hop distance for anchor nodes. The shuffled frog leaping (SFL) algorithm [46], [47] offers the advantages of fewer parameters, faster calculation, and stronger optimization for WSNs. SFL simulates the information exchange process, implemented in the classification of subgroups using a fitness function, as virtual frogs search for food. This approach combines a global search with subgroup interior searching, prompting the algorithm to proceed in a globally optimal direction. The key is to design a fitness function capable of solving application-specific scenarios, as discussed below.

The average hop distance for each anchor node (i) can be expressed as:

$$AvgDis_i = \frac{\sum_{j=1}^{N_i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{\sum_{j=1}^{N_i} h_{ij}}, \tag{21}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) are the positions of anchor nodes i and j, respectively. The term h_{ij} is the hop count between anchor nodes and N_i is the number of anchors connected by node i. The distance estimation between the unknown node k and the anchor node i is given by $AvgDis_i \times h_{ik}$, where h_{ik} is the hop count between nodes k and i. The actual distance can be expressed as:

$$d_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}.$$

The difference between these two terms is represented by:

$$e_{ik} = |d_{ik} - AvgDis_i \times h_{ik}|, \tag{22}$$

where h_{ik} is the minimum hop count between unknown node k and the anchor node i. Smaller values of the error e_{ik}

result in more accurate average hop distances ($AvgDis_i$). Therefore, the solution to (22) can be transformed into the following minimum value optimization problem:

$$f(AvgDis_i) = \frac{1}{N_k - 1} \sum\nolimits_{i \neq k}^{N_k} e_{ik}. \tag{23}$$

Different weights must be considered for each node in order to reduce hop distance errors. In FIGURE 16, the actual distance between the anchor nodes (A or B) and P is approximately the signal radius R. However, the hop count between A and P is 1 and the hop count between B and P is 2. Thus, the hop distance between P and B is 2*R*. As such, the hop distance error will increase significantly if the fitness value for anchors A and B is the same.

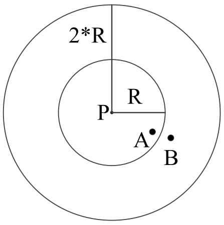


FIGURE 16. Weighted values for different nodes.

A modified fitness function can then be expressed as:

$$f(AvgDis_i) = \frac{1}{N_k - 1} \sum\nolimits_{i \neq k}^{N_k} a^2 e_{ik}^2, \quad a = \frac{d_{ik}}{h_{ik}}, \tag{24}$$

where N_k is the number of anchor nodes, d_{ik} is the actual distance between anchor nodes, and h_{ik} is the hop count between anchor nodes. This new fitness function was used in conjunction with the improved frog leaping algorithm to solve for optimal average hop distance.

Step3: Position estimation

Distance estimation errors typically increase with hop count. However, estimated positions for target nodes were restricted to smaller regions using the rough localization, to prevent estimation from being included in too many hop counts. Due to the complexity of the algorithm to be reduced, a least squares optimization technique was applied to further calculate unknown node positions within the small region [48].

The distance between target nodes and the anchor node (i) is given by $d_{ik} = AvgDis_i \times h_{ik}$. The corresponding position estimation is then:

$$X = (A^T A)^{-1} A^T B, \tag{25}$$

where

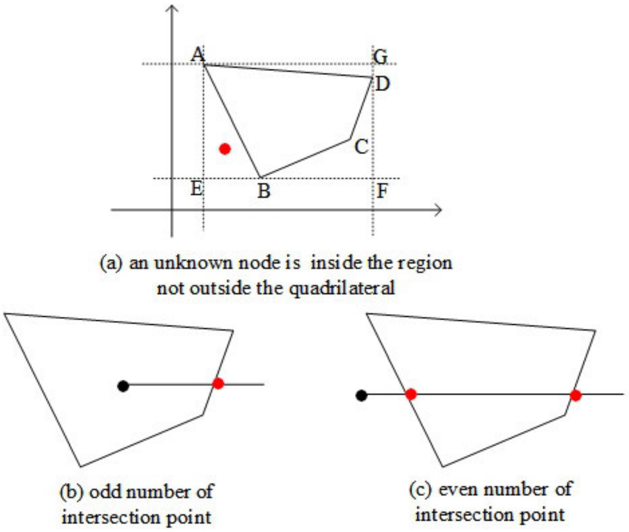
$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{26}$$

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) & 2(z_1 - z_n) \\ \vdots & \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2\,(y_{n-1} - y_n) & 2(z_{n-1} - z_n) \end{bmatrix}, \tag{27}$$

$$B = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + z_1^2 - z_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + z_{n-1}^2 - z_n^2 + d_{n-1}^2 - d_1^2 \end{bmatrix}. \tag{28}$$

3) FUSION CALIBRATION

The QSC algorithm is based on RSSI values and the DVMFL algorithm is based on hop counts. Each model has unique advantages for estimating position. Therefore, we propose a fusion calibration framework, combining both algorithms, to further improve localization precision. The QSC algorithm produces larger position estimation errors for unknown nodes lying outside the quadrilateral region or anchor nodes that do not form a quadrilateral. As such, two judgment rules were used to determine whether an unknown node was located within a quadrilateral, as shown in FIGURE 17.



(a) an unknown node is inside the region not outside the quadrilateral

(b) odd number of intersection point

(c) even number of intersection point

FIGURE 17. Various types of intersection points. (a) An unknown node is inside the region but not outside the quadrilateral. (b) An odd number of intersection points. (c) An even number of intersection points.

Rule #1: Minimum and maximum coordinates are searched within the quadrilateral vertices. If an unknown node is not located within this range, the node is assumed to lie outside the quadrilateral. This process is demonstrated in FIGURE 17(a).

Rule #2: If the node does lie inside a region composed of four vertices (A,G,E, and F) (i.e., Rule #1 is satisfied), intersections between unknown node rays and quadrilateral edges can be used for further assessment. Specifically, if the number of intersecting points is odd, the node lies inside the quadrilateral.

The fusion calibration formula can then be expressed as:

$$\alpha \times QSC + \beta \times DVMFL, \tag{29}$$

where α is a weighting factor in the QSC algorithm, β is a weighting factor in the DVMFL algorithm, and $\beta = 1 - \alpha$.

Algorithm 3 A Judgement to Determine Whether an Unknown Node Is Inside or Outside a Quadrilateral

```
Input: An unknown node $(x_{UN}, y_{UN}, z_{UN})$ and an anchor
       node $(x^i_{AN}, y^i_{AN}, z^i_{AN})$, i ∈ 1, 2, 3, 4
       $x_{max}, x_{min} \leftarrow \max(x^i_{AN})$ and $\min(x^i_{AN_i})$
       $y_{max}, y_{min} \leftarrow \max(y^i_{AN})$ and $\min(y^i_{AN})$
if $(x_{min} \leq x \leq x_{max})$ and $(y_{min} \leq y \leq y_{max})$ then:
  return True
else
  P = False
    for $i = 1$ to 4, $j = 4$ to$l$ do:
      if $((y_i > y)! = (y_j > y))$ and $(x < ((x_j - x_i)(y - y_i))/(y_j - y_i) + x_i)$
      P = !P
      end if
    end for
    returnP
end if
Output: Unknown node is either inside or outside the quadrilateral
```

In this study, the algorithm was used to test 200 unknown nodes. When it was determined that a node was positioned within a quadrilateral area composed of cooperative anchor nodes, the weight α was set to different values for comparison. These results are shown in the table below. The ALE was minimized, indicating the highest position estimation accuracy, when $\alpha = 0.8$, as shown in Table 2. As such, the value of α was set to 0.8.

TABLE 2. ALE corresponding to different α values.

α	ALE
0.5	0.0476
0.6	0.0471
0.7	0.0463
0.8	0.0456
0.9	0.0458
1.0	0.0461

The value of α will be larger ($\alpha = 0.8$ in this study) if Rules #1 and #2 apply, otherwise β will be larger. The values of α and β were determined through empirical testing. This process is represented in Algorithm 3.

IV. EXPERIMENTAL EVALUATION

This section analyzes the performance of our hybrid model, considering different parameters and various conditions, such as anchor node density and network topology. The proposed technique is also compared to conventional methods.

A. SIMULATION CONFIGURATIONS

All experiments were conducted on a PC with a Windows 7 (64-bit) operation system, running an Intel Core i5-6300 2.33 GHz CPU with 20 GB DDR-SDRAM memory and a 1 TB hard drive. All code was developed in Python 3.6.2 using the Spyder 3.3.4 IDE.

For the sake of simplicity, the testing process was performed in a 100 m × 100 m × 100 m cubic region in which the anchor and unknown nodes were randomly positioned. The number of nodes and related parameters were developed empirically as shown in Table 3. Ultimately, 200 unknown nodes were used to evaluate location estimation accuracy. Table 3 provides a detailed summary of the experimental configuration used to assess the proposed algorithm.

The training process for the MLGB model is described below.

Step 1: A higher learning velocity is selected (set $\nu = 0.1$) and the default values for other parameters are established: $m_d = -1$, $S = 1$, $F = 1$, $\lambda = 0$, $\gamma = 0$, and $T = 31$. The runtime for 300 iterations is acquired.

Step 2: Basic parameters are adjusted in the decision tree φ_T, a vector consisting of $m_d = [4, 6, 8, 10]$ and min-split-gain = 0. A grid search is then used to fine-tune the parameters in a specific range.

Step 3: The regularization parameters λ and γ are optimized.

Step 4: The learning velocity is reduced and the number of iterations is increased to further optimize model parameters.

Specific parameter values after adjustment and optimization are shown in Table 3.

Non-uniform deployment was also investigated using C-shape, H-shape, S-shape, and O-shape networks. The effects of node density and sensing coverage were specifically tested. The number of nodes was varied from 0 to 200, producing different point densities. The coverage (i.e., the communication radius) was varied from 20 m to 80 m using a 10 m step size.

In this manner, the training and test datasets are produced under the same simulation conditions. A total of 10000 and 200 recorded points were acquired after data processing and feature extraction, respectively, for both datasets.

Localization performance was quantified using the average location estimation (ALE) error defined in the (29). The ALE metric used in the experiment is similar to that of previous studies [17], but with slight modifications. It is given by:

$$ALE = \frac{\sum_{i=1}^{n} \|P_{est} - P_{sensor}\|^2}{n \cdot R}, \tag{30}$$

where $P_{est} = [x'_{i\_est}\ y'_{i\_est}\ z'_{i\_est}]^T$ is the estimated position of the i^{th} unknown node, $P_{sensor} = [x_i\ y_i\ z_i]^T$ is the actual position of the i^{th} unknown node, n is the total number of unknown nodes in the sensing coverage region, and R is the signal coverage radius between nodes. Smaller ALE values indicate higher location estimation precision.

TABLE 3. Experimental parameter configuration.

Parameter	Definition	Value
B_n	Number of anchor nodes	25-50
T_n	Number of training samples	10000
U_n	Number of unknown nodes	100-200
R	Communication radius	20-80 m
D	Sensor network region	100 × 100 × 100
d_0/m	RSSI reference distance	1
σ^2	Gaussian random noise variance	2
δ	Error threshold	0.2
θ	Rotation angle	60º
ν	Learning velocity	0.05
m_d	Maximum tree depth	6
S	Sample sampling ratio	0.8
F	Feature sampling ratio	0.5
λ	Regularization parameters	1.6
γ	Regularization parameters	2.5
T	Number of leaf nodes	54

B. VARYING ANCHOR NODE DEPLOYMENT

The effect of varying node density on location estimation error was investigated experimentally. In this process, 25 anchor nodes with signal radii ranging from 20-80m were randomly distributed in the sample space. The results for 40 and 80 m coverage distances are plotted in FIGURE 18. As shown in Table 4, the ALE for this configuration was 0.0464 with random deployment. This value was only 8.2% higher than the result produced with uniform deployment. In other words, anchor node distribution had little effect on ALE errors.

We further evaluated location estimation errors at the center and edges of the sensing range for uneven distributions. A 90 × 90 × 90 m simulated area was considered the center of the coverage region; the remaining area was marginal. Consequently, we derive that 104 nodes were assigned to the marginal area and 96 nodes were placed in the center region by counting nodes in different region.

FIGURE 19 shows a series of simulation results in which there was little difference between positioning errors at the center and edges of the sensing coverage region. As shown in Table 5, the max-min error in the marginal and center regions was 0.1236 and 0.0816 for randomly distributed anchor nodes, respectively. This suggests the positioning errors are well distributed in the sensing coverage region, allowing the proposed algorithm to compensate in marginal areas.

TABLE 4. Simulation results from four network topologies.

Network Topology	Maximum Error	Minimum Error	Average Error	Median	Standard Deviation
C-shape	0.2473	0.0059	0.0628	0.0467	0.0459
H-shape	0.2918	0.0031	0.0512	0.0413	0.0388
S-shape	0.3208	0.0077	0.0514	0.0415	0.0402
O-shape	0.2651	0.0061	0.0517	0.0440	0.0350

TABLE 5. ALE results for varying anchor node distributions.

Deployment Mode	Ave error (Total)	Ave error (Center)	Max-Min error (Center)	Ave error (Margin)	Max-Min error (Margin)
Uniform	0.0429	0.0290	0.0827	0.0465	0.1163
Random	0.0464	0.0296	0.0816	0.0456	0.1236

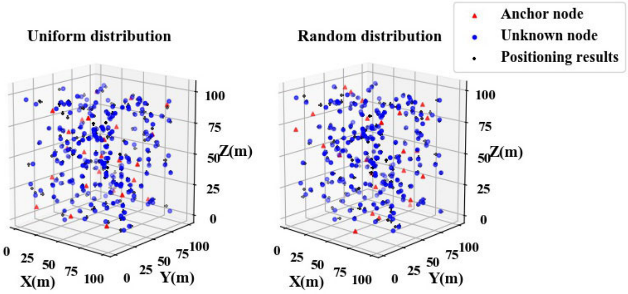


FIGURE 18. The results of uniform and random anchor node deployment strategies.

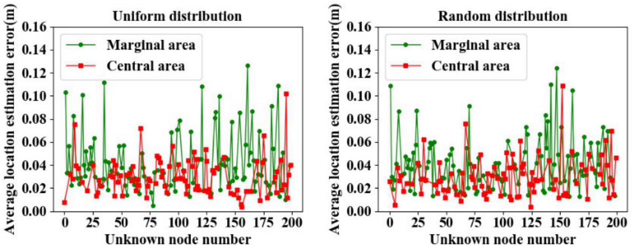


FIGURE 19. A comparison of center and edge regions.

C. RESULTS FOR DIFFERING NETWORK TOPOLOGIES

An experiment was conducted to investigate non-uniform deployment scenarios in four distinct network topologies. In practical applications, node distributions are not uniform and are restricted to specific network architectures. As such, C-shape, H-shape, S-shape, and O-shape networks were studied, to assess the effects of positioning accuracy with 25 anchor nodes. The communication radius in this simulation ranged from 30 to 80 m, as shown in FIGURE 20.

The influence of location estimation errors with four different network shapes is also evident in the figure. The ALE was lowest in the ‘H-shape’ network and highest in the ‘C-shape’ network. Detailed simulation results are also provided in Table 4. The ALE values were 8.9% higher than those of Table 5 (random deployment), with a maximum difference of 33.6%. As such, the effect of network topology on location estimation accuracy is evident.

FIGURE 21 shows the probability density distribution for location estimation errors in four network topologies. The primary component of each curve is inclined to the left by a factor of 0.34, which indicates the majority of positioning

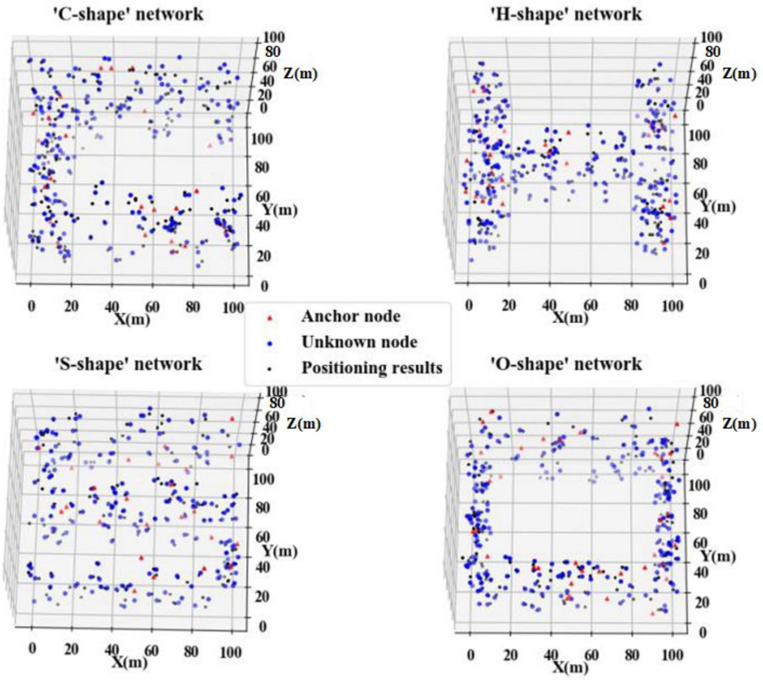


FIGURE 20. Experimental results from four network topologies.

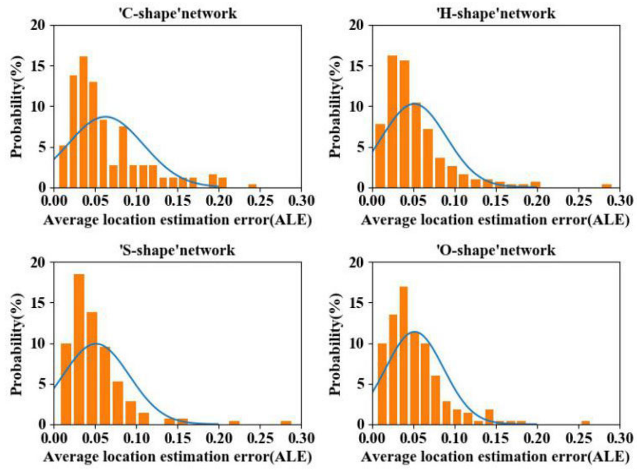


FIGURE 21. A graph of error distributions from four network topologies.

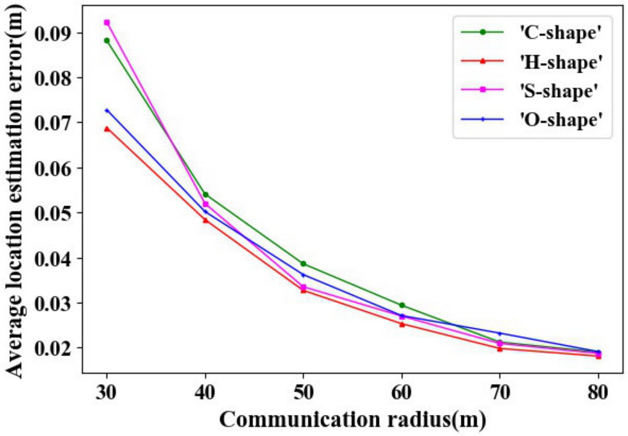


FIGURE 22. The effect of signal radius on location estimation errors in four network topologies.

errors are smaller (less than the mean value). The shape of each curve is similar to that of a normal distribution. For example, in the ‘C-shape’ network, the area of $\mu + 2\sigma$ (the sum of ALE values and double the standard deviation) accounted for 95% of the area.

This represents the proportion of nodes (4.8%) in which the location estimation errors were above $\mu + 2\sigma$ were. This suggests the probability of large errors is very small and that ALE was relatively uniform and low. These results suggest our algorithm is robust and offers improved localization performance for a variety of network topologies.

As shown in FIGURE 22, the ALE across four network shapes decreased with increasing communication radius. The curve also decreased rapidly when the communication radius was less than 50 m. The error gap between the four network shapes also tended to decrease gradually as the location estimation error approached 80 m. The largest change was observed in the ‘S-shape’ network, in which the location error was reduced by 79.7%. The trend in the ‘H-shape’ structure was similar to that of the ‘O-shape’ network and the influence of signal radius was the smallest among network types. From this, we conclude that the impact of different network shapes on location estimation accuracy is relatively small for the proposed algorithm.

D. ROUGH LOCALIZATION SIMULATION RESULTS

In this section, the performance of MLGB is evaluated and compared with four conventional algorithms. The effect of

RSSI data sparsity on location estimation errors was evaluated experimentally. Tests were performed over 100 × 100 × 100 m cubic areas with 25 anchor nodes randomly distributed over a sensing coverage distance of 40 m. Four different models including Random forest [16], GBDT [41], XGBoost [49], and LightGBM [40] were used to assess the performance of rough localization in the first stage of the proposed algorithm.

FIGURE 23 shows a series of simulation results in which the percentage of missed values (RSSI sparsity) directly affected location estimation errors using different models. The average location estimation error was high when the percent of missed values was high, increasing significantly as the percentage rose from 50% to 60%. It is also evident that the GBDT model produced the lowest accuracy, while the proposed model outperformed the other four algorithms. Our technique offers improved sparse feature processing capabilities and a new fitness function for WSN localization.

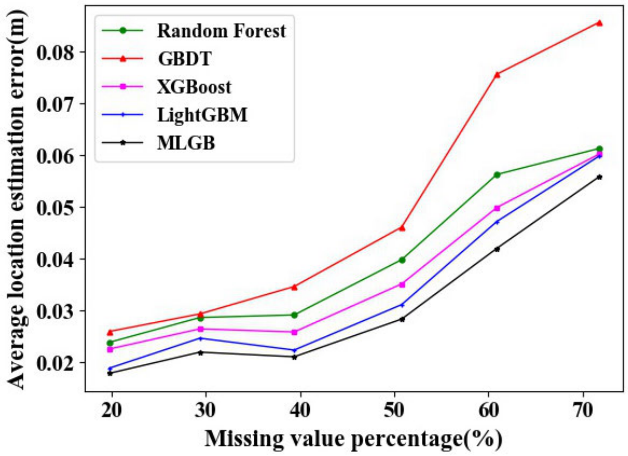


FIGURE 23. The effect of RSSI data sparsity on location estimation errors.

E. EXPERIMENTAL RESULTS FOR SCREENING COOPERATIVE NODE STRATEGIES

Screening cooperative node strategies were evaluated in the second stage of the algorithm. An experiment was performed using the same configuration as in Section IV, in which the results were compared to those not utilizing a screening strategy. This comparison included four nodes with the nearest distance for cooperative anchor nodes. The results are shown in Table 6.

FIGURE 24 shows the average location estimation errors produced with different strategies. It is evident from these results that the effect of screening on ALE is relatively low, while the accuracy exhibits minor improvements. The largest improvement in prediction accuracy was observed along the X-axis, due to the two corrections occurring in the X-axis direction. The three-dimensional ALE was as much as 14.8% lower with screening than without, suggesting the strategy smallest location estimation errors for all algorithms were observed as the percentage of anchor nodes increased, particularly from 20-50%. The proposed algorithm achieved the lowest estimation errors, which remained mostly constant, fluctuating only slightly as the anchor node density passed 30%. This is because an increase in the percentage of anchor nodes leads to an increase in data dimensionality. Higher-dimensional features can provide more information and additional missing improved overall precision.

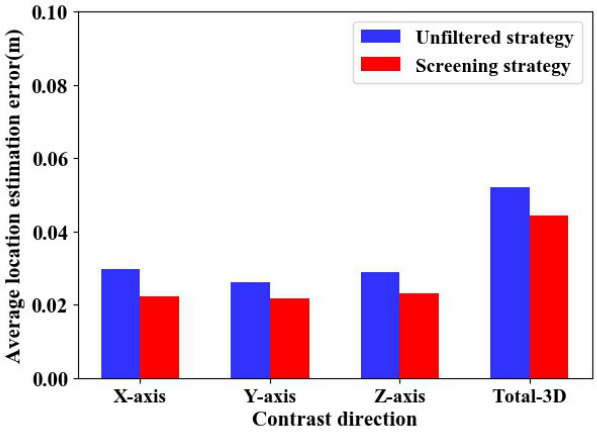


FIGURE 24. A comparison of different localization strategies.

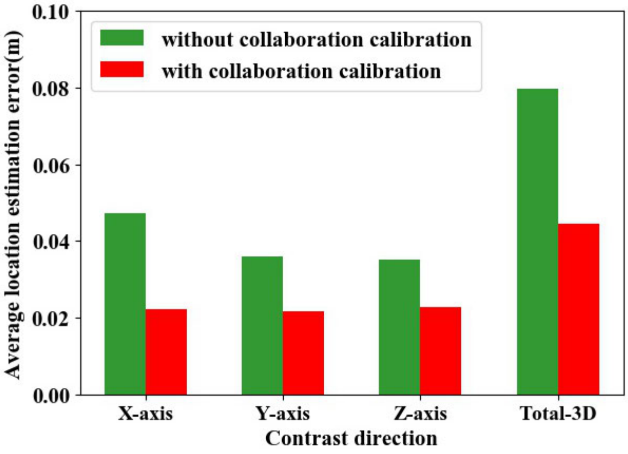


FIGURE 25. A comparison of results with and without collaboration calibration.

F. SIMULATION RESULTS FOR COLLABORATIVE FUSION CALIBRATION

In this section, we evaluate performance in the third stage of our approach. Position coordinates for unknown nodes acquired by rough localization were further calibrated using the QSC and DVMFL algorithms, which are based on anchor nodes (including screening cooperative anchor nodes). The effectiveness of collaborative fusion fine calibration was evaluated using 3D positioning errors along the X-, Y-, and Z- axes, as shown in Table 7.

FIGURE 25 provides a comparison of results produced with and without collaboration calibration. The collaborative fusion algorithm reduced positioning errors along the X, Y, and Z-axes. While errors in each of the three directions were similar, precision in the X direction showed the largest improvement, decreasing positioning errors by 53% (as a

TABLE 6. A comparison of results using screening and unfiltered strategies.

		Maximum Error	Minimum Error	Average Error	Median	Standard Deviation
Unfiltered Strategy	X-axis	0.1293	0.0001	0.0297	0.0216	0.0165
	Y-axis	0.1127	0.0001	0.0261	0.0193	0.0174
	Z-axis	0.1680	0.0002	0.0288	0.0213	0.0188
	Total-3D	0.1731	0.0046	0.0522	0.0475	0.0231
Screening Strategy	X-axis	0.0952	0.0000	0.0222	0.0186	0.0179
	Y-axis	0.0907	0.0000	0.0217	0.0155	0.0197
	Z-axis	0.1406	0.0000	0.0230	0.0187	0.0201
	Total-3D	0.1489	0.0039	0.0445	0.0380	0.0249

TABLE 7. Location errors along the X-, Y-, and Z-axEs, with 200 unknown nodes.

		Maximum Error	Minimum Error	Average Error	Median	Standard Deviation
Prior to Collaboration	X-axis	0.8590	0.0001	0.0473	0.0266	0.0875
	Y-axis	0.2686	0.0002	0.0359	0.0243	0.0379
	Z-axis	0.2901	0.0001	0.0351	0.0212	0.0425
	Total-3D	0.8838	0.0055	0.0795	0.0532	0.0969
After the Collaboration	X-axis	0.0952	0.0000	0.0222	0.0186	0.0179
	Y-axis	0.0907	0.0000	0.0217	0.0155	0.0197
	Z-axis	0.1406	0.0000	0.0230	0.0187	0.0201
	Total-3D	0.1489	0.0039	0.0445	0.0380	0.0249

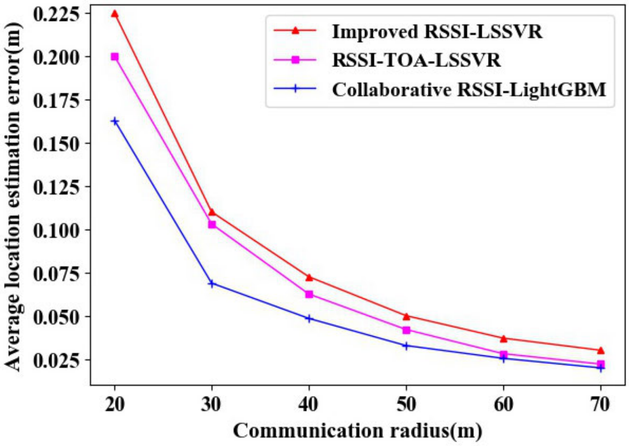


FIGURE 26. The effect of communication radius on location estimation error.

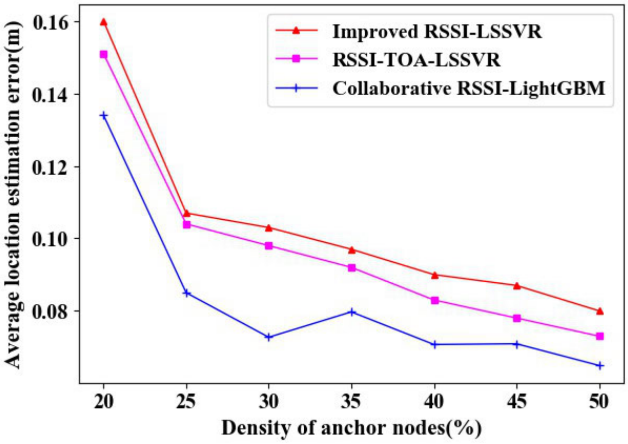


FIGURE 27. The effect of anchor node density on location estimation error.

result of two corrections in the X direction). The total average location estimation error in three dimensions was also reduced by 44.0%.

G. A COMPARISON OF CONVENTIONAL TECHNIQUES

A comparative evaluation of the whole algorithm was performed using the improved RSSI-LSSVR (least squares support vector regression) [50] and RSSI-TOA-LSSVR [51] algorithms, which were based on a machine-learning model and applied to node localization in 3D environments. In addition, the configuration shown in Table 3 was used to investigate three factors: communication radius, anchor node density, and ranging-distance error, all of which can effect location estimation accuracy.

FIGURE 26 shows location estimation errors with communication radii ranging from 20 to 70 m. These errors decreased with increasing communication radius, particularly over the 20-30 m range, before stabilizing over larger values. The proposed algorithm outperformed other models in the same conditions, producing the smallest and most consistent positioning errors, indicating our model was the least affected by communication radius.

FIGURE 27 shows location estimation errors for anchor node percentages ranging from 20-50%, with other conditions unchanged. This error increased with node density. The values. Enhancing data sparsity can also lead to an increase in positioning errors.

FIGURE 28 shows the error for measurement distances ranging from 0.5-3.0 m, with other conditions held constant. The difference between measured and real distances increased with range distance, as did the maximum location estimation error. The superior performance offered by our

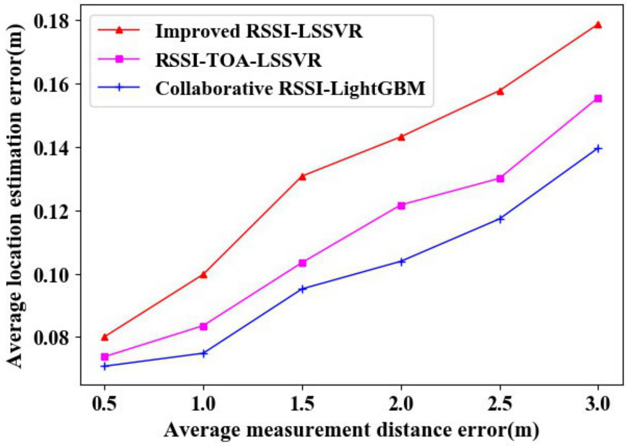


FIGURE 28. The effect of range distance error on location estimation error.

algorithm is a result of adopting a dynamic signal propagation model, which can effectively reduce environmental effects on measurement distances. The proposed model was also used to establish a relationship between signal feature space and position coordinates in a rough localization stage, rather than relying solely on range information, which reduced the effects of range distance errors.

V. CONCLUSION

In this study, a hybrid approach was proposed and applied to localization approximations for WSNs in 3D environments. Specifically, the effects of uneven localization error distributions, RSSI data sparsity, and irregular network topology were investigated, including their effects on localization estimation. Our contribution to this novel algorithm included a modified version of the LightGBM, QSC, and DVMFL algorithms, as well as a new screening cooperative anchor node policy with a multistage collaborative calibration scheme.

This process included three primary components. First, MLGB-based rough localization was used to estimate the positions of unknown nodes by overcoming issues that affected localization precision. Second, a cooperative technique was used to improve anchor node density. Third, QSC and DVML models were combined to establish a collaborative fusion calibration framework to improve location estimation precision. Experimental evaluation was conducted by varying factors to be considered include such as node distribution, non-uniform error distribution, network topology, signal radius, and RSSI data sparsity. Results showed that the proposed model outperformed conventional techniques. we intend to continue developing applications of this work in several engineering fields in a future study.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for suggestions and insights that helped to improve the study. They would also like to thank LetPub (www.letpub.com) for providing linguistic assistance during the preparation of this manuscript.

REFERENCES

[1] Z. Wang, X. Wang, L. Liu, M. Huang, and Y. Zhang, ‘‘Decentralized feedback control for wireless sensor and actuator networks with multiple controllers,’’ *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 5, pp. 1471–1483, Oct. 2017, doi: 10.1007/s13042-016-0518-y.

[2] S. B. Chandanapalli, E. S. Reddy, and D. R. Lakshmi, ‘‘DFTDT: Distributed functional tangent decision tree for aqua status prediction in wireless sensor networks,’’ *Int. J. Mach. Learn. Cybern.*, vol. 3, pp. 1419–1434, Mar. 2017, doi: 10.1007/s13042-017-0653-0.

[3] H. Alemdar and C. Ersoy, ‘‘Wireless sensor networks for healthcare: A survey,’’ *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, Oct. 2010, doi: 10.1109/jproc.2010.2065210.

[4] R. S. Alonso, I. Sittón-Candanedo, Ó. García, J. Prieto, S. Rodríguez-González, ‘‘An intelligent Edge-IoT platform for monitoring livestock and crops in a dairy farming scenario,’’ *Ad Hoc Netw.*, vol. 98, Mar. 2020, Art. no. 102047, doi: 10.1016/j.adhoc.2019.102047.

[5] J. P. Dominguez-Morales, A. Rios-Navarro, M. Dominguez-Morales, R. Tapiador-Morales, D. Gutierrez-Galan, D. Cascado-Caballero, A. Jimenez-Fernandez, and A. Linares-Barranco, ‘‘Wireless sensor network for wildlife tracking and behavior classification of animals in Doñana,’’ *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2534–2537, Dec. 2016, doi: 10.1109/.LCOMM.2016.2612652.

[6] M. Erdelj, M. Król, and E. Natalizio, ‘‘Wireless sensor networks and multi-UAV systems for natural disaster management,’’ *Comput. Netw.*, vol. 124, pp. 72–86, Sep. 2017, doi: 10.1016/j.comnet.2017.05.021.

[7] A. J. AL-Mousawi and H. K. AL-Hassani, ‘‘A survey in wireless sensor network for explosives detection,’’ *Comput. Elect. Eng.*, vol. 72, pp. 682–701, Nov. 2018, doi: 10.1016/j.compeleceng.2017.11.013.

[8] M. ur Rahman, S. Rahman, S. Mansoor, V. Deep, and M. Aashkaar, ‘‘Implementation of ICT and wireless sensor networks for earthquake alert and disaster management in earthquake prone areas,’’ *Procedia Comput. Sci.*, vol. 85, pp. 92–99, May 2016, doi: 10.1016/j.procs.2016.05.184.

[9] Y. Zhang, W. Wu, and Y. Chen, ‘‘A range-based localization algorithm for wireless sensor networks,’’ *J. Commun. Netw.*, vol. 7, no. 4, pp. 429–437, Dec. 2005, doi: 10.1109/jcn.2005.6387985.

[10] L. Cui, C. Xu, G. Li, Z. Ming, Y. Feng, and N. Lu, ‘‘A high accurate localization algorithm with DV-hop and differential evolution for wireless sensor network,’’ *Appl. Soft Comput.*, vol. 68, pp. 39–52, Jul. 2018.

[11] S. Tomic and I. Mezei, ‘‘Improvements of DV-hop localization algorithm for wireless sensor networks,’’ *Telecommun. Syst.*, vol. 61, no. 1, pp. 93–106, Mar. 2015, doi: 10.1007/s11235-015-0014-9.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, ‘‘Range-free localization schemes for large scale sensor networks,’’ in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2003, pp. 81–95.

[13] J. Kuriakose, S. Joshi, R. V. Raju, and A. Kilaru, ‘‘A review on localization in wireless sensor networks,’’ in *Advances in Signal Processing and Intelligent Recognition Systems* (Advances in Intelligent Systems and Computing), S. Thampi, A. Gelbukh, and J. Mukhopadhyay, Eds. Cham, Switzerland: Springer, 2014, pp. 599–610, doi: 10.1007/978-3-319-04960-1_52.

[14] P. Cottone, S. Gaglio, G. L. Re, and M. Ortolani, ‘‘A machine learning approach for user localization exploiting connectivity data,’’ *Eng. Appl. Artif. Intell.*, vol. 50, pp. 125–134, Apr. 2016, doi: 10.1016/j.engappai.2015.12.015.

[15] F. Zhu and J. Wei, ‘‘Localization algorithm for large scale wireless sensor networks based on fast-SVM,’’ *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 1859–1875, Sep. 2016, doi: 10.1007/s11277-016-3665-2.

[16] B. Alotaibi and K. Elleithy, ‘‘A new MAC address spoofing detection technique based on random forests,’’ *Sensors*, vol. 16, no.3, p. 281, Mar. 2016, doi: 10.3390/s16030281.

[17] S. K. Gharghan, R. Nordin, M. Ismail, and J. A. Ali, ‘‘Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling,’’ *IEEE Sensors J.*, vol. 16, no. 2, pp. 529–541, Jan. 2016, doi: 10.1109/JSEN.2015.2483745.

[18] S. Phoemphon, C. So-In, and D. (Tao) Niyato, ‘‘A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization,’’ *Appl. Soft Comput.*, vol. 65, pp. 101–120, Apr. 2018, doi: 10.1016/j.asoc.2018.01.004.

[19] Z. Wang, H. Zhang, T. Lu, Y. Sun, and X. Liu, ‘‘A new range-free localisation in wireless sensor networks using support vector machine,’’ *Int. J. Electron.*, vol. 105, no. 2, pp. 244–261, Jul. 2017, doi: 10.1080/00207217.2017.1357198.

[20] Y. Guo, B. Sun, N. Li, and D. Fang, ‘‘Variational Bayesian inference-based counting and localization for off-grid targets with faulty prior information in wireless sensor networks,’’ *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1273–1283, Mar. 2018, doi: 10.1109/TCOMM.2017.2770139.

[21] X. Li, S. Ding, and Y. Li, ''Outlier suppression via non-convex robust PCA for efficient localization in wireless sensor networks,'' *IEEE Sensors J.*, vol. 17, no. 21, pp. 7053–7063, Nov. 2017, doi: 10.1109/JSEN.2017.2754502.

[22] J. Yoo and H. Kim, ''Target localization in wireless sensor networks using online semi-supervised support vector regression,'' *Sensors*, vol. 15, no. 6, pp. 12539–12559, May 2015, doi: 10.3390/s150612539.

[23] T. J. Chowdhury, C. Elkin, V. Devabhaktuni, D. B. Rawat, and J. Oluoch, ''Advances on localization techniques for wireless sensor networks: A survey,'' *Comput. Netw.*, vol. 110, pp. 284–305, Dec. 2016, doi: 10.1016/j.comnet.2016.10.006.

[24] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple, ''A survey of localization in wireless sensor network,'' *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 12, Dec. 2012, Art. no. 962523, doi: 10.1155/2013/457874.

[25] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, ''Machine learning algorithms for wireless sensor networks: A survey,'' *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019, doi: 10.1016/j.inffus.2018.09.013.

[26] M. Bernas and B. Płaczek, ''Fully connected neural networks ensemble with signal strength clustering for indoor localization in wireless sensor networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 12, Dec. 2015, Art. no. 403242, doi: 10.1155/2015/403242.

[27] S. S. Banihashemian, F. Adibnia, and M. A. Sarram, ''A new range-free and storage-efficient localization algorithm using neural networks in wireless sensor networks,'' *Wireless Pers. Commun.*, vol. 98, no. 1, pp. 1547–1568, Jan. 2018.

[28] S. Yun, J. Lee, W. Chung, and E. Kim, ''Centroid localization method in wireless sensor networks using TSK fuzzy modeling,'' in *Proc. Int. Symp. Adv. Intell. Syst.*, vol. 2005, pp. 971–974.

[29] S. Amri, F. Khelifi, A. Bradai, A. Rachedi, and M. L. Kaddachi, ''A new fuzzy logic based node localization mechanism for wireless sensor networks,'' *Future Gener. Comput. Syst.*, vol. 93, pp. 799–813, Apr. 2019, doi: 10.1016/j.future.2017.10.023.

[30] S. Kumar, S. N. Tiwari, and R. M. Hegde, ''Sensor node tracking using semi-supervised Hidden Markov Models,'' *Ad Hoc Netw.*, vol. 33, pp. 55–70, Oct. 2015, doi: 10.1016/j.adhoc.2015.04.004.

[31] S. Chang, ''A deep learning approach for localization systems of high-speed objects,'' *IEEE Access*, vol. 7, pp. 96521–96530, 2019, doi: 10.1109/ACCESS.2019.2929444.

[32] H. Wymeersch, J. Lien, and M. Z. Win, ''Cooperative localization in wireless networks,'' *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009, doi: 10.1109/JPROC.2008.2008853.

[33] A. Stanoev, S. Filiposk, V. In, and L. Kocarev, ''Cooperative method for wireless sensor network localization,'' *Ad Hoc Netw.*, vol. 40, pp. 61–72, Apr. 2016, doi: 10.1016/j.adhoc.2016.01.003.

[34] E. Hu, Z. L. Deng, M. Hu, L. Yin, and W. Liu, ''Cooperative indoor positioning with factor graph based on FIM for wireless sensor network,'' *Future Gener. Comput. Syst.*, vol. 89, pp. 126–136, Dec. 2018, doi: 10.1016./j.future.2018.05.035.

[35] A. Kumar, A. Khosla, J. S. Saini, and S. Singh, ''Stochastic algorithms for 3D node localization in anisotropic wireless sensor networks,'' in *Proc. 7th Int. Conf. Bio-Inspired Comput., Theories Appl. (BIC-TA)*. Springer, 2013, pp. 1–14.

[36] A. Kumara, A. Khosla, J. S. Saini, and S. S. Sidhu, ''Range-free 3D node localization in anisotropic wireless sensor networks,'' *Appl. Soft Comput.*, vol. 34, pp. 438–448, Sep. 2015, doi: 10.1016/j.asoc.2015.05.025.

[37] R. Preeth, R. Mohan, and V. Balakrishnan, ''Localization in wireless sensor networks: A dimension based pruning approach in 3D environments,'' *Appl. Soft Comput.*, vol. 68, pp. 219–232, Jul. 2018, doi: 10.1016/j.asoc.2018.03.039.

[38] F. Zhu and J. Wei, ''Localization algorithm for large-scale wireless sensor networks based on FCMTSR-support vector machine,'' *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 10, Oct. 2016, Art. no. 1550147716674010, doi: 10.1177/1550147716674010.

[39] S. Phoemphon, C. So-In, and N. Leelathakul, ''A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks,'' *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113044.

[40] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ''LightGBM: A highly efficient gradient boosting decision tree,'' in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 1–8.

[41] J. H. Friedman, ''Greedy function approximation: A gradient boosting machine,'' *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001. [Online]. Available: https://projecteuclid.org/euclid.aos/1013203451, doi: 10.1214/aos/1013203451.

[42] J. Sander, ''LOF: Identifying density-based local outliers,'' *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000, doi: 10.1145/342009.335388.

[43] S. Tomic, M. Beko, and R. Dinis, ''RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes,'' *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2037–2050, May 2015.

[44] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, ''Cooperative received signal strength-based sensor localization with unknown transmit powers,'' *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, Mar. 2013, doi: 10.1109/TSP.2012.2232664.

[45] S. Gu, Y. Yue, C. Maple, and C. Wu, ''Fuzzy logic based localisation in wireless sensor networks for disaster environments,'' in *Proc. 18th Int. Conf. Automat. Comput. (ICAC)*, Sep. 2012, pp. 1–5.

[46] M. Muzaffar and K. E. Lansey, ''Water distribution network design using the shuffled frog leaping algorithm,'' *J. Water Resour. Planning Manage.*, vol. 129, no. 3, pp. 210–225, May 2003, doi: 10.1061/40569(2001)412.

[47] B. Hu, Y. Dai, Y. Su, P. Moore, X. Zhang, C. Mao, J. Chen, and L. Xu, ''Feature selection for optimized high-dimensional biomedical data using an improved shuffled frog leaping algorithm,'' *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1765–1773, Nov. 2018.

[48] R. Lin, B. Hu, and W. J. Shi, ''Incremental multi-hop localization algorithm based on regularized weighted least squares,'' *Int. J. Pattern Recognit. Artif. Intell.*, vol. 33, no. 9, Mar. 2019, Art. no. 1959032, doi: 10.1142/S0218001419590328.

[49] M. Luckner, B. Topolski, and M. Mazurek, ''Application of XGBoost algorithm in fingerprinting localisation task,'' in *Proc. 16th IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.*, Bialystok, Poland, Jun. 2017, pp. 16–18.

[50] L. Zhang, H. Yang, Y. Yu, and F. Peng, ''A three-dimensional node security localization method for WSN based on improved RSSI-LSSVR algorithm,'' in *Proc. 10th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Feb. 2018, pp. 182–186.

[51] L. Zhang, Z. Wang, Z. Kuang, and H. Yang, ''Three-dimensional localization algorithm for WSN nodes based on RSSI-TOA and LSSVR method,'' in *Proc. 11th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Apr. 2019, pp. 498–503.

LIANG XU (Member, IEEE) received the master's degree in control theory and control engineering from the Lanzhou University of Technology, in 2002, and the Ph.D. degree in computer science and technology from the South China University of Technology, in 2007. He is currently an Associate Professor with the School of Automation, Guangdong University of Technology. His current research interests include machine vision, wireless sensor networks, and deep learning.

ZHILIANG LI received the B.E. degree in electrical engineering and automation from the Guangdong University of Petrochemical Technology, in 2016. He is currently pursuing the master's degree in control engineering with the School of Automation, Guangdong University of Technology. His research interests include wireless sensor networks, machine learning, and deep learning.

XIUXI LI received the master's and Ph.D. degrees in chemical engineering from the South China University of Technology, in 1998 and 2008, respectively. He is currently a Professor with the School of Chemical and Chemical Engineering, South China University of Technology. His current research interests include chemical process safety, product quality control, and process optimal.